トラブルシューティングのツールと方法論

コマンドラインインターフェイスのトラブルシューティング コマンド

コマンドラインインターフェイス（CLI）を使用すると、ローカルコンソールを使用して、またはTelnetまたはセキュアシェル（SSH）セッションを使用してリモートで設定およびモニタできます。Cisco NX-OSCLIには、Cisco IOSソフトウェアに似たコマンド構造があり、状況依存ヘルプ、**show**コマンド、マルチユーザ サポート、およびロールベースのアクセス制御が備わっています。

各機能には、機能の設定、ステータス、パフォーマンスに関する情報を提供する**show**コマンドが用意されています。また、次のコマンドを使用すると、さらに詳しい情報を確認することができます。

- **show system**コア、エラー、および例外を含むシステムレベルのコンポーネントに関する情報を提供します。**show system error-id**コマンドを使用し、コマンドにより、エラーコードの詳細を検索できます。

```
switch# copy running-config startup-config
[########################################] 100%
2013 May 16 09:59:29 zoom %$ VDC-1 %$ %BOOTVAR-2-AUTOCOPY_FAILED: Autocopy of file
/bootflash/n9000-dk9.6.1.2.I1.1.bin to standby

switch# show system error-id 0x401e0008
Error Facility:     sysmgr
Error Description:  request was aborted, standby disk may be full
```

整合性チェッカー コマンド

Cisco NX-OSには、ソフトウェア状態とハードウェア状態を検証する整合性チェッカー コマンドが用意されています。整合性チェッカーの結果は、PASSEDまたはFAILEDとして記録されます。

```
2019 May 1 16:31:39 switch vshd: CC_LINK_STATE:
Consistency Check: PASSED
```

整合性チェッカーは、次の機能を実行するツールです。

- システムの整合性を確認する
- 根本原因分析と障害分離の実行を支援する
- ソフトウェア テーブルとハードウェア テーブル間の整合性をチェックする

（注） モニターセッションがダウン状態またはエラー状態の場合、整合性チェッカーは検証されません。

Cisco NX-OS は、次の整合性チェッカーをサポートします。

表 *1*:整合性チェッカー コマンド

コマンド	説明	サポートされるプラットフォーム
show consistency-checker copp	CoPPプログラミングを確認します。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FX、-Rラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ Cisco Nexus N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GXスイッチ
show consistency-checker copp extended module *module_no* [**brief** | **detail**[ ] （注） Cisco NX-OS リリース 10.5（3）F以降、このコマンドは廃止されました。	すべてのコントロールプレーンACL の整合性を確認します。 • **[概要（Brief）]**：失敗したエントリの出力を構造化形式で表示します。 **[詳細（Detail）]**：すべてのACEエントリの出力を構造化形式で表示します。	Cisco Nexus 9300-FX3/GX/GX2/H2R/H1、9808、および9804 シリーズ スイッチ。
show consistency-checker control-plane acl extended module *module_no* [**brief** | **detail**]	すべてのコントロールプレーンACL の整合性を確認します。この新しいコマンドは、CoPPチェックをより効果的に実行するために修飾子が増えて拡張されています。	Cisco Nexus 9300-FX/FX2/FX3/GX/GX2/H2R/H1、9808 および9804 シリーズ スイッチおよび、9700-EX/FX/FX3/GX ライン カード付きの9500 シリーズ スイッチ
show consistency-checker dme interfaces	DMEインターフェイスを確認します。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FX ライン カードを備えたCisco Nexus 9500プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker egress-xlate private-vlan	ハードウェアのプライベート VLAN egress-xlate を確認します。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ
show consistency-checker fex-interfaces {**fex** *fex-id* | **interface ethernet** *fex-id/fex-slot/fex-port*} [**brief** | **detail**]	FEX インターフェイスのソフトウェアとハードウェアの状態を比較します。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ （注） *fex-slot* は常に 1 です。
show consistency-checker fex-interfaces fabric <*fabric-po*>	物理メンバー インターフェイスの FEX ファブリック PO メンバーシップ、およびファブリック ポート チャネル メンバーのインターフェイス レベルのハードウェア プログラミングを確認します。	Cisco Nexus 9300-EX、9300-FX、9300-FX2 および 9300-GX シリーズ スイッチ。
show consistency-checker fex-interfaces fabric <*fabric-po*> **membership vlan** <*vlan-id*>	FEX インターフェイスで有効になっている VLAN について、FEX ファブリック PO メンバーが VLAN フラッドリストの一部であることを確認します。	Cisco Nexus 9300-EX、9300-FX、9300-FX2 および 9300-GX シリーズ スイッチ。
show consistency-checker fex-interfaces fabric <*fabric-po*> **stp-state vlan** <*vlan-id*>	FEX インターフェイスで有効になっている VLAN の FEX ファブリック PO メンバーが転送/無効状態であることを確認します。	Cisco Nexus 9300-EX、9300-FX、9300-FX2 および 9300-GX シリーズ スイッチ。
show consistency-checker fex-interfaces fabric <*fabric-po*> **egress-xlate private-vlan** <*vlan-id*>	PVLAN 対応の FEX インターフェイスがある場合に、FEX ファブリック PO インターフェイスに対応する PVLAN ハードウェア プログラミングを確認します。	Cisco Nexus 9300-EX、9300-FX、9300-FX2 および 9300-GX シリーズ スイッチ。

コマンド	説明	サポートされるプラットフォーム
test consistency-checker forwarding {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]	レイヤ 3 整合性チェッカーを開始します。	Cisco Nexus 9000 シリーズ スイッチ
show consistency-checker forwarding {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]	レイヤ 3 整合性チェッカーテスト結果を表示します。	すべての Cisco Nexus 9000 シリーズ スイッチ
show consistency-checker forwarding single-route {**ipv4** | **ipv6**} *ip-address* **vrf** *vrf-name*} [**brief** | **detail**]	特定のルートのレイヤ 3 ルートの整合性をチェックします。ECMP グループ テーブルの枯渇が原因で単一ルートが失敗したときに警告します。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチ （注） Cisco Nexus 34180YC プラットフォーム スイッチでは、**ipv4** コマンドのみをサポートしています。
show consistency-checker gwmacdb	ゲートウェイ MAC アドレス データベースのハードウェアとソフトウェアの一貫性をチェックします。 （注） このコマンドは、4 ウェイ HSRP に対して誤った結果を表示する場合があります。	すべての Cisco Nexus 9000 シリーズ スイッチ
show consistency-checker kim interface {**ethernet** *slot/port* | **port-channel** *number* | **vlan** *vlan-id*} [**brief** | **detail**]	スーパーバイザとライン カード間の内部接続を確認します。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker l2 module *module-number*	学習した MAC アドレスがソフトウェアとハードウェア間で一貫していることを確認します。また、ハードウェアに存在するがソフトウェアには存在しない追加エントリと、ハードウェアに存在しないエントリも表示されます。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker l2 multicast group *ip-address* **source** *ip-address* **vlan** *vlan-id* [**brief** | **detail**]	レイヤ 2 マルチキャスト グループとの不整合をチェックします。	Cisco Nexus 9200、9300-EX、9300-FX、および9300-GX プラットフォーム スイッチおよび Cisco Nexus 9500 プラットフォーム スイッチ-EX および -FXライン カード N9K-X9432C-S、N9K-X9536PQ ライン カード搭載の Cisco Nexus 9500 シリーズ スイッチ N9K-X9432C-FM-S、N9K-C9508-FMX-S、N9K-C9508-FM-S ファブリック モジュールを搭載した Cisco Nexus 9500 シリーズ スイッチ。 Cisco Nexus N3K-C3232C、N3K-C3264Q、N3K-C31108TC-V、N3K-C3132Q-40GX、N3K-C3132Q-V、N3K-C31108PC-V、N3K-C3172PQ、N3K-C3172TQ、N3K-C3164Q、および N3K-C3164Q -10GE スイッチ。 Cisco Nexus N9K-C9372TX、N9K-C9372TX-E、N9K-C93120TX、N9K-X9432C-S、N9K-C9332PQ、N9K-C9372PX、および N9K-C9372PX-E スイッチ。
show consistency-checker l2 switchport interface {**ethernet** *slot/port* | **port-channel** *number* }[**brief** | **detail** | **all**]	スイッチポートインターフェイスとの不整合をチェックします。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker l3-interface interface ethernet *slot/port* [**brief** | **detail**]	ハードウェアのインターフェイスのレイヤ 3 設定と、ハードウェアの L3VLAN、CMLフラグ、IPv4イネーブル、VPN IDの設定を確認します。このコマンドは、物理インターフェイスおよびポートチャネルの一部であるインターフェイスに対して機能します。サブインターフェイスまたはFEXインターフェイスは検証されません。 Cisco NX-OS リリース 9.3(5) 以降、このコマンドは SI および SVI インターフェイスのレイヤ 3 設定をチェックします。サポートは Cisco Nexus 9300-GX プラットフォーム スイッチにも拡張されます。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチ Cisco Nexus N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX デバイス。 Cisco NX-OS リリース 10.3(1)F 以降、L3 整合性チェッカは Cisco Nexus 9808 プラットフォーム スイッチでサポートされています。 Cisco NX-OS リリース 10.4（1）F 以降、L3 一貫性チェッカーは、Cisco Nexus 9808 スイッチ（Cisco Nexus X98900CD-A、X9836DM-A ライン カード搭載）サポートされます。 Cisco NX-OS リリース 10.4（1）F 以降、L3 一貫性チェッカーは Cisco Nexus 9804 プラットフォーム スイッチ、Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。 Cisco NX-OS リリース 10.4（2）F 以降、L3 整合性チェッカーは Cisco Nexus C9232E-B1 スイッチでサポートされます。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker l3-interface module *module-number* [**brief** | **detail**]	モジュール内のすべてのインターフェイスのレイヤ 3 設定と、ハードウェアの L3VLAN、CML フラグ、IPv4 イネーブル、VPN ID の設定を確認します。このコマンドは、物理インターフェイスおよびポートチャネルの一部であるインターフェイスに対して機能します。サブインターフェイスは検証されません。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker l3 multicast group *ip-address* **source** *ip-address* **vrf** *vrf-name* [**brief** | **detail**]	レイヤ3 マルチキャスト グループとの不整合をチェックします。	Cisco Nexus 9200、9300-EX、9300-FX、および9300-GX プラットフォーム スイッチおよび Cisco Nexus 9500 プラットフォーム スイッチ-EX および -FXライン カード N9K-X9432C-S、N9K-X9536PQ ライン カードを搭載した Cisco Nexus 9500 シリーズスイッチ、および N9K-X9432C-FM-S、N9K-C9508-FMX-S、および N9K-C9508-FM-S ファブリック モジュール。 Cisco Nexus N3K-C3048TP、N3K-C3064-TC、N3K-C3232C、N3K-C3264Q、N3K-C31108TC-V、N3K-C3132Q-40GX、N3K-C3132Q-V、N3K-C31108PC-V、N3K-C3172PQ、C3172TQ、N3K-C3164Q、および N3K-C31128PQ-10GE スイッチ。 Cisco Nexus N9K-C9372TX、N9K-C9372TX-E、N9K-C93120TX、N9K-X9432C-S、N9K-C9332PQ、N9K-C9372PX、および N9K-C9372PX-E スイッチ。
show consistency-checker link-state fabric-ieth [**module** *module-number*] [**brief** | **detail**]	内部ファブリック ポートのリンク状態ステータスについて、ソフトウェアとハードウェア間のプログラミングの一貫性を確認します。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker link-state interface ethernet *slot/port* [**brief** | **detail**]	インターフェイスのリンク状態ステータスについて、ソフトウェアとハードウェア間のプログラミングの一貫性を確認します。このコマンドは、物理イーサネットインターフェイスおよびポートチャネルの一部である物理イーサネットインターフェイスに対して機能します。サブインターフェイスまたはFEXインターフェイスは検証されません。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FXラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ
show consistency-checker link-state module *module-number* [**brief** | **detail**]	モジュール内のすべてのインターフェイスのソフトウェア リンク状態をハードウェア リンク状態と照合します。このコマンドは、物理イーサネット インターフェイスおよびポートチャネルの一部である物理イーサネットインターフェイスに対して機能します。サブインターフェイスまたは FEX インターフェイスは検証されません。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FX、-Rラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ
show consistency-checker membership port-channels [**interface port-channel** *channel-number*] [**brief** | **detail**]	すべてのモジュールのハードウェアのポート チャネル メンバーシップをチェックし、ソフトウェア状態で検証します。このコマンドは、ポートチャネルごとに実行されます。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FX、-Rラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ
show consistency-checker membership port-channels [**brief** | **detail**]	すべてのモジュールのハードウェアのポート チャネル メンバーシップをチェックし、ソフトウェア状態で検証します。このコマンドは、システム内のすべてのポートチャネルに対して実行されます。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FX、-Rラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker membership vlan *vlan-id* {**native-vlan** | **private-vlan interface** {**ethernet** *slot/port* | **port-channel** *number* | **native-vlan**}} [**brief** | **detail** | **interface**]	ソフトウェアのVLANメンバーシップがハードウェアにプログラムされているものと同じであることを判別します。また、STP BLK状態のインターフェイスも無視します。	Cisco Nexus 9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FX、-Rラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ （注） **private-vlan** コマンドでの **brief** または **detail** オプションはサポートされていません。 （注） Cisco Nexus 34180YCプラットフォーム スイッチでは、**native-vlan** コマンドのみをサポートしています。
show consistency-checker pacl {**module** *module-number* | **port-channels interface port-channel** *channel-number*}	ハードウェアとソフトウェア間のIPv4、IPv6、およびMAC PACLプログラミングの整合性を検証し、<label, entry-location>ペアはハードウェアとソフトウェアの間で一貫しています。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FXラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ
show consistency-checker pacl extended ingress {**ip** | **ipv6** | **mac**} **interface** {**ethernet** *slot/port* | **port-channel** *number*} [**brief** | **detail**]	入力インターフェイス（FEXインターフェイスを含む）およびポート チャネルのPACLプログラミングを確認します。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FXラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ
show consistency-checker pacl extended ingress {**ip** | **ipv6** | **mac**} **module** *module-number* [**brief** | **detail**]	指定されたモジュールのすべての物理インターフェイス、サブインターフェイス、ブレークアウト ポート、およびFEXインターフェイスでPACLプログラミングを確認します。	Cisco Nexus 34180YC、9200、9300-EX、および9300-FXプラットフォーム スイッチ、および-EX、-FXラインカードを備えたCisco Nexus 9500プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker port-state fabric-ieth [**module** *module-number* [**ieth-port** *ieth-port*]] [**brief** | **detail**]	内部ファブリック ポートの状態を確認します。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ
show consistency-checker port-state [**module** *module-number*] [**brief** | **detail**]	指定されたモジュールのポートの状態を確認します。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker racl {**module** *module-number* | **port-channels interface port-channel** *channel-number* | **svi interface vlan** *vlan-id*}	ハードウェアとソフトウェア間の IPv4 および IPv6 RACL プログラミングの一貫性を検証し、<label, entry-location> ペアはハードウェアとソフトウェアの間で一貫しています。 • このコマンドは、モジュールごとに呼び出されると、そのモジュールのすべての物理インターフェイスおよびサブインターフェイスの IPv4 および IPv6 ACL の整合性を確認します。 • 特定のポートチャネルでこのコマンドを呼び出すと、すべてのメンバーポートが検証されます。 • すべてのポートチャネルでこのコマンドを呼び出すと、このコマンドはACLが適用されているポートチャネルごとに確認します。 （注） このコマンドは、IPv4 および IPv6 ACL を検証せず、修飾子とアクションが一致するかどうかを検証しません。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ
show consistency-checker racl extended ingress {**ip** | **ipv6**} **interface** {**ethernet** *slot/port* | **port-channel** *number* | **vlan** *vlan-id*} [**brief** | **detail**]	入力インターフェイス、サブインターフェイス、ブレークアウトポート、ポートチャネル、またはSVIのRACLプログラミングを確認します。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker racl extended ingress {**ip** | **ipv6**} **module** *module-number* [**brief** | **detail**]	指定されたモジュールの入力インターフェイスの RACL プログラミングを確認します。このコマンドは、そのモジュールのすべての物理インターフェイス、サブインターフェイス、およびブレークアウト ポートで実行されます。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ
show consistency-checker stp-state vlan *vlan-id* [**brief** | **detail** | **interface**]	ソフトウェアのスパニング ツリーの状態が、ハードウェアでプログラミングされた状態と同じかどうかを判別します。このコマンドは、動作中（アップ）のインターフェイスでのみ実行されます。	Cisco Nexus34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチおよび -EX、-FX、および -R ライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチ。
show consistency-checker vacl extended ingress {**ip** | **ipv6** | **mac**} **vlan** *vlan-id* [**brief** | **detail**]	VLAN のすべてのメンバー インターフェイスで VACL プログラミングを確認します。	Cisco Nexus 34180YC、9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ

コマンド	説明	サポートされるプラットフォーム
show consistency-checker vpc [**source-interface**] [**brief** | **detail**]	vPC の不整合をチェックします。出力マスクを持たないポートの LACP 個別（I）状態を確認します。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ N9K-X9432C-S、N9K-X9536PQ ライン カードを搭載した Cisco Nexus 9500 シリーズスイッチ、および N9K-X9432C-FM-S、N9K-C9508-FMX-S、および N9K-C9508-FM-S ファブリック モジュール。 Cisco Nexus N3K-C3048TP、N3K-C3064-TC、N3K-C3232C、N3K-C3264Q、N3K-C31108TC-V、N3K-C3132Q-40GX、N3K-C3132Q-V、N3K-C31108PC-V、N3K-C3172PQ、C3172TQ、N3K-C3164Q、および N3K-C31128PQ-10GE スイッチ。 Cisco Nexus N9K-C9372TX、N9K-C9372TX-E、N9K-C93120TX、N9K-X9432C-S、N9K-C9332PQ、N9K-C9372PX、および N9K-C9372PX-E スイッチ。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker vxlan config-check [**verbose-mode**]	スイッチの VXLAN EVPN 設定を確認します。	Cisco Nexus 9200、9300-EX および 9300-FX プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。 Cisco Nexus C3132Q-40GE-SUP、C3132Q-40GX-SUP、C3132Q-XL、C31128PQ-10GE、C3264Q-S、C3264C-E スイッチ。
show consistency-checker vxlan infra [**verbose-mode**]	VXLAN トンネル インフラストラクチャとの不整合をチェックします。	Cisco Nexus 9200、9300-EX および 9300-FX プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。 Cisco Nexus C3132Q-40GE-SUP、C3132Q-40GX-SUP、C3132Q-XL、C31128PQ-10GE、C3264Q-S、C3264C-E スイッチ。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker vxlan l2 module *module-number*	VXLAN レイヤ 2 ルートとの整合性を確認します。	Cisco Nexus 9200、9300-EX および 9300-FX プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。 Cisco Nexus C3132Q-40GE-SUP、C3132Q-40GX-SUP、C3132Q-XL、C31128PQ-10GE、C3264Q-S、C3264C-E スイッチ。
show consistency-checker vxlan l3 vrf [*vrf-name* | **all**] [**start-scan** | **report**]	VXLAN レイヤ 3 ルートとの不一致をチェックします。	Cisco Nexus 9200、9300-EX および 9300-FX プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。
show consistency-checker vxlan pv	ソフトウェア間およびハードウェアの異なるテーブル間で VLAN マッピングが一貫してプログラムされているかどうかを確認します。このコマンドを実行するには、少なくとも 1 つのインターフェイスでポート VLANマッピングを有効にする必要があります。	Cisco Nexus 9200、9300-EX および 9300-FX/FX2 および 9500 プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker vxlan qinq-qinvni	ソフトウェアおよびハードウェアで一貫しているマルチタグ VLAN リストおよび関連するマルチタグ vn-segment をチェックします。	Cisco Nexus 9300-FX/FX2 プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。
show consistency-checker vxlan selective-qinvni interface {**ethernet** *slot/port* | **port-channel** *channel-number*}	パケット内の内部タグが保持されるように、ポート固有の選択的 Q-in-VNI マッピングがソフトウェアおよびハードウェアで正しくプログラムされているかどうかを検証します。	Cisco Nexus 9300-EX および 9300-FX/FX2 プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。
show consistency-checker vxlan vlan [**all** | *vlan-id*] [**verbose-mode**]	VXLAN VLAN との不一致をチェックします。	Cisco Nexus 9300-EX および 9300-FX/FX2 プラットフォーム スイッチ Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX、および X9536PQ スイッチ。 Cisco Nexus C3132Q-40GE-SUP、C3132Q-40GX-SUP、C3132Q-XL、C31128PQ-10GE、C3264Q-S、C3264C-E スイッチ。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker tap-aggregation qinq	ポート tap-aggregation および qinq との不整合をチェックします。	Cisco Nexus N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX、N9K-C9504-FM-G、and N9KC9508- FM-G スイッチおよび N9K-X9716D-GX ライン カード
show consistency-checker vxlan xconnect	VXLAN Xconnect VLAN との不一致をチェックします。Xconnect ACL がすべてのユニットとスライスにインストールされ、MAC 学習がすべての Xconnect VLAN で無効になっていることを検証します。	Cisco Nexus 9200、9332C、9364C、9300-EX、および 9300-FX/FX2 プラットフォーム スイッチ。
show consistency-checker vxlan l3 single-route [**ipv4** | **ipv6**] [*vrf*]	VXLAN レイヤ 3 シングル ルート トラフィックとの不整合をチェックします。	Cisco Nexus 9200、9300-EX および 9300-FX プラットフォーム スイッチ。 Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX および X9536PQ スイッチ、Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker vxlan l2 [**mac-address**] [*mac-address*] | **module**] [*module*	VXLAN レイヤ 2 との不一致をチェックします。	Cisco Nexus 9200、9300-EX および 9300-FX プラットフォーム スイッチ。 Cisco Nexus C31108PC-V、C31108TC-V、C3132Q-V、および 3132C-Z スイッチ。 Cisco Nexus C9396TX、C93128TX、C9396PX、X9564PX、X9564TX および X9536PQ スイッチ、Cisco Nexus 9200、9300-EX、および9300-FXプラットフォームスイッチ。 Cisco Nexus C3132Q-40GE-SUP、C3132Q-40GX-SUP、C3132Q-XL、C31128PQ-10GE、C3264Q-S、C3264C-E スイッチ。

コマンド	説明	サポートされるプラットフォーム
show consistency-checker storm-control [**brief** | **detail**]	ストーム制御との不整合をチェック	

コマンド	説明	サポートされるプラットフォーム
		Cisco NX-OS リリース 10.5（1）以降、ストーム制御の一貫性の概要と詳細。 Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX、-R ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ Cisco NX-OSリリース9.3(5)以降では、N3K-C3016Q-40GE、N3K-C3048TP-1GE、N3K-C3064PQ-10GE、N3K-C3064PQ-10GX、N3K-C3064T-10GT、N9K-C9504- FM、N9K-C9508-FM、N9K-C9516-FM、N9K-C9508-FM-S、N3K-C31128PQ、N3K-C3164Q-40GE、N3K-C3232C、N3K-C3132Q-V、N3K-C31108PC-V、N3K-C31108P-V C31108TC-V、N3K-C3264C-E、N3K-C3132C-Z、N9K-C93128TX、N9K-C9396PX、N9K-C9372PX、および N9K-C9332PQ デバイスでサポートされています。 （注） ND ISSU が Cisco NX-OS リリース 10 に対して実行されます。4（x）であり、ハードウェアとソフトウェアの pol_rate または pol_burst 値が一致しない場合、ストーム

コマンド	説明	サポートされるプラットフォーム
		制御整合性チェッカーは失敗します。この問題を解決するには、ストーム制御を再構成します。
show consistency-checker segment-routing mpls [**ip**] [*ip-address*] | **mask**] [*mask* | **vrf**] [*vrf*	アンダーレイ セグメント ルーティング（ISIS、BGP、OSPF）およびレイヤ 3 VPN およびレイヤ 2 EVPN オーバーレイ ルートのルート整合性をチェックします。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ。 Cisco Nexus N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX デバイス。
show consistency-checker segment-routing mpls label	アンダーレイ セグメント ルーティング（ISIS、BGP、OSPF）およびオーバーレイ ルートのレイヤ 3 VPN、レイヤ 2 EVPN、および ADJ SIDS のラベル整合性をチェックします。	Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチ、および -EX、-FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチ。 Cisco Nexus N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX デバイス。
show consistency-checker sflow [**brief** | **detail**]	スーパーバイザーとライン カード ハードウェア テーブルのプログラムと整合性構成をチェックします。	Cisco Nexus 9300-FX2、9300-FX3、9300-GX および 9300-GX2 プラットフォーム スイッチ （注） Cisco NX-OS リリース 10.3（3）F 以降、Cisco Nexus 9808 プラットフォーム スイッチです

次のコマンドは JSON 出力をサポートしていません。

- **show consistency-checker forwarding** {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]

- **show consistency-checker pacl** {**module** *module-number* | **port-channels interface port-channel** *channel-number*}
- **show consistency-checker racl module** *module-number*
- **show consistency-checker racl port-channels interface port-channel** *channel-number*}
- **show consistency-checker racl svi interface vlan** *vlan-id*
- **show consistency-checker vxlan**
- **test consistency-checker forwarding** {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]

show consistency-checker vxlan コマンドはモデル化されていません。

マルチキャスト整合性チェッカー

マルチキャスト整合性チェッカーは、マルチキャストルートの状態を確認するためのレイヤ 2 およびレイヤ 3 ルートの単一ルート整合性チェッカーです。マルチキャスト整合性チェッカーは、各コンポーネントで show コマンドを実行し、関連情報を解析し、処理された情報を他のコンポーネントと比較して不整合をチェックします。マルチキャスト整合性チェッカー コマンドは、障害が発生すると終了します。**show consistency-checker l2 multicast group** および **show consistency-checker l3 multicast group** コマンドは、期待値と実際の値の差を返します。

これらのコマンドは、次の出力形式をサポートしています。

- **verbose**：結果をテキスト形式で表示します。
- **detail**：結果を JSON 形式で表示します。
- **brief**：結果を最小限の詳細とともに JSON 形式で表示します。

Cisco NX-OS リリース 10.2(2)F 以降、L3 マルチキャスト整合性チェッカーは NAT 変換をサポートし、すべてのプラットフォームでサポートされています。UMNAT はサポートされていません。

（注） MMNAT は Multicast to Multicast NAT を表し、MUNAT は Multicast to Unicast NAT を表し、UMNAT は Unicast to Multicast NAT を表します。NAT 変換は、タイプ MMNAT 入力および出力、および MUNAT である必要があります。

Cisco NX-OSリリース10.2(1)F 以降では、Multicast over GRE 整合性チェッカーが N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX ファミリ スイッチに導入されています。Multicast over GRE（mGRE）整合性チェッカは次をサポートしています。

- シングル ルートS mGRE 整合性チェッカ
- L3イーサネットインターフェイス、L3ポートチャネル、およびL3サブインターフェイス上のmGREトンネル

- トランスポートプロトコルVRFがトンネルインターフェイスVRFと異なる場合があるGREトンネル。これは、GREv4-IPv4マルチキャストを介したGREトンネルでのみサポートされます。

Multicast over GRE（mGRE）整合性チェッカは次をサポートしていません。

- FEX
- IPv6を介したGREトンネル
- mGREはEoRではサポートされていません。整合性チェックは、N9K-C9316D-GX、N9KC93600CD-GX、N9K-C9364C-GX ToRでのみサポートされます。
- mGRE は SVI ではサポートされていません。

mGRE整合性チェックは、発信インターフェイスリストにIP GREトンネルインターフェイスがある場合、またはRPFインターフェイスがIP GREトンネルインターフェイスである場合にのみ実行されます。

Cisco NX-OSリリース10.1(1) 以降では、次の整合性チェッカーがサポートされています。

- IPv6 L2 マルチキャスト整合性チェッカー
- IPv6 L3 マルチキャスト整合性チェッカー
- マルチキャスト NLB 整合性チェッカー
  - マルチキャスト MAC ルックアップ モード整合性チェッカー
  - マルチキャスト NLB L3 ユニキャスト設定整合性チェッカー
- マルチキャスト GRE 整合性チェッカー

次の既存の CLI コマンドは、IPv6 L2 マルチキャスト整合性チェッカーの IPv6 送信元およびグループ アドレスを受け入れるように拡張されています。

show consistency-checker l2 multicast group <ipv4/ipv6 group address> source <ipv4/v6 source address> vrf <vrf-id> [brief|detail]

次に、IPv6 L2 マルチキャスト整合性チェッカーの出力例を示します。

```
# show consistency-checker l2 multicast group ?
  A.B.C.D   Group IP address
  A:B::C:D  Group IPv6 address
```

次の既存の CLI コマンドは、IPv6 L3 マルチキャスト整合性チェッカーの IPv6 送信元およびグループ アドレスを受け入れるように拡張されています。

show consistency-checker l3 multicast group <ipv4/ipv6 group address> source <ipv4/v6 source address> vlan <vlan-id> [brief|detail]

次に、IPv6 L3 マルチキャスト整合性チェッカーの出力例を示します。

```
# show consistency-checker l3 multicast group ?
  A.B.C.D   Group IP address
  A:B::C:D  Group IPv6 address
```

マルチキャストMACルックアップモードの整合性チェッカーをサポートするために、次の新しいCLIコマンドが追加されました。

show consistency-checker l2 multicast mac <mac> vlan <vlan-id>

次に、マルチキャストMACルックアップモードの整合性チェッカーの出力例を示します。

```
# show consistency-checker l2 multicast mac 0100.1234.1234 vlan 10 ?
  >       Redirect it to a file
  >>      Redirect it to a file in append mode
  brief   Show consistency checker structured output in brief
  detail  Show consistency checker structured output in detail
  |       Pipe command output to filter
```

（注） このCLIは、MACルックアップモードの整合性チェッカまたはNLBのL2モードの整合性チェッカーに使用されます。入力MACは、ip-macまたはnon-ip-macのいずれかです。

マルチキャストNLB L3ユニキャスト設定整合性チェッカーをサポートするために、次の新しいCLIコマンドが追加されました。

show consistency-checker multicast nlb cluster-ip <unicast-cluster-ip> vrf <vrf-id>

次に、マルチキャストNLB L3ユニキャスト設定整合性チェッカーの出力例を示します。

```
# show consistency-checker multicast nlb cluster-ip <unicast-cluster-ip>
  >       Redirect it to a file
  >>      Redirect it to a file in append mode
  brief   Show consistency checker structured output in brief
  detail  Show consistency checker structured output in detail
  |       Pipe command output to filter
```

次の既存のCLIコマンドは、マルチキャストGRE整合性チェッカーに使用されます。

show consistency-checker l3 multicast group <ipv4 group address> source <ipv4 source address> vrf <vrf-id> [brief|detail]

（注） 既存のIPv4 L3マルチキャスト整合性チェッカーCLIを使用して、マルチキャストGRE整合性チェッカーを開始します。

マルチキャスト整合性チェッカーは、次のデバイスをサポートしています。

- Cisco Nexus 92304QC、9272Q、9232C、9236C、92300YC、93108TC-EX、93180LC-EX、93180YC-EX、and 9300-GXプラットフォームスイッチおよびN9K-X9736C-EX、N9K-X97160YC-EX、N9K-X9732C-EX、およびN9K-X9732C-EXMラインカードです。
- N9K-X96136YC-R、N9K-X9636C-R、およびN9K-X9636Q-Rラインカードを搭載したCisco Nexus 9500シリーズスイッチ。

Cisco NX-OS Release 9.3(5)以降では、マルチキャスト整合性チェッカーは次のデバイスをサポートしています。

- N9K-X9432C-S、N9K-X9536PQ ライン カードを搭載した Cisco Nexus 9500 シリーズスイッチ、および N9K-X9432C-FM-S、N9K-C9508-FMX-S、および N9K-C9508-FM-S ファブリック モジュール。
- Cisco Nexus N3K-C3048TP、N3K-C3064-TC、N3K-C3232C、N3K-C3264Q、N3K-C31108TC-V、N3K-C3132Q-40GX、N3K-C3132Q-V、N3K-C31108PC-V、N3K-C3172PQ、N3K-C3172TQ、N3K-C3164Q、および N3K-C31128PQ-10GE スイッチ。
- Cisco Nexus N9K-C9372TX、N9K-C9372TX-E、N9K-C93120TX、N9K-X9432C-S、N9K-C9332PQ、N9K-C9372PX、および N9K-C9372PX-スイッチ。

Cisco NX-OSリリース10.1(1) 以降では、マルチキャスト整合性チェッカーは次のデバイスをサポートしています。

- Cisco Nexus N9k-C9504 を搭載した N9K-X97160YC-EX、N9k-C9504 を搭載した N9K-X9732C-EX、N9k-C9504 を搭載した N9K-X9732C-FX、N9k-C9504 を搭載した N9K-X9736C-EX、N9k-C9504 を搭載した N9K-X9736C-FX、N9k-C9504 を搭載した N9K-X9736Q-FX、および N9k-C9504 を搭載した N9K-X9788TC-FX。
- Cisco Nexus N9k-C9508 を搭載した N9K-X97160YC-EX、N9k-C9508 を搭載した N9K-X9732C-EX、N9k-C9508 を搭載した N9K-X9732C-FX、N9k-C9508 を搭載した N9K-X9736C-EX、N9k-C9508 を搭載した N9K-X9736C-FX、N9k-C9508 を搭載した N9K-X9736Q-FX、および N9k-C9508 を搭載した N9K-X9788TC-FX。
- Cisco NX-OS リリース 10.3（1）F 以降、マルチキャスト整合性チェッカーは Cisco Nexus 9808 プラットフォーム スイッチでサポートされています。
  - Cisco NX-OS リリース 10.4（1）F 以降、マルチキャスト一貫性チェッカーは、Cisco Nexus 9808 スイッチ（Cisco Nexus X98900CD-A、X9836DM-A ライン カード搭載）サポートされます。

Cisco NX-OS リリース 10.4（1）F 以降、マルチキャスト一貫性チェッカーは、Cisco Nexus 9804 プラットフォーム スイッチ（Cisco Nexus X98900CD-A、X9836DM-A ライン カード搭載）でサポートされます。

Cisco NX-OS リリース 10.4（2）F 以降、マルチキャスト整合性チェッカーは Cisco Nexus 9232E-B1 プラットフォーム スイッチでサポートされます。

マルチキャスト整合性チェッカーは、次のレイヤ2コンポーネントのプログラミングの整合性を検証します：

- IGMP スヌーピング
- MFDM
- MFIBPI
- MFIBPD
- ハードウェア テーブル

マルチキャスト整合性チェッカーは、次のレイヤ3コンポーネントのプログラミングの整合性を検証します：

- PIM
- MRIB
- IGMP スヌーピング
- MFDM
- MFIBPI
- MFIBPD
- ハードウェア テーブル

Cisco NX-OS リリース 10.5（3）F 以降、レイヤー 3 整合性チェッカは Cisco Nexus N9364E-SG2-Q プラットフォーム スイッチでサポートされています。

マルチキャスト整合性チェッカ コマンドの出力例

次に、IGMP スヌーピングの出力例を示します。

```
switch# show ip igmp snooping groups 225.12.12.28 225.12.12.28 vlan 222
Type: S - Static, D - Dynamic, R - Router port, F - Fabricpath core port
Vlan  Group Address      Ver  Type  Port list
222   225.12.12.28       v3   D     Eth1/2 Eth1/3 Po12 Po100 Po18
```

次に、MFDM の出力例を示します。

```
switch# show forwarding distribution l2 multicast vlan 222 group 225.12.12.28 source
225.12.12.28
Vlan: 222, Group: 225.12.12.28, Source: 225.12.12.28
  Outgoing Interface List Index: 4
  Reference Count: 204
  Num L3 usages: 4
  Platform Index: 0xa00004
  Vpc peer link exclude flag set
  Number of Outgoing Interfaces: 5
    Ethernet1/2
    Ethernet1/3
    port-channel12
    port-channel18
    port-channel100
```

次に、IGMP スヌーピングと MFDM を比較する例（成功）を示します。

```
**********************************************************************
Comparing IGMP Snooping with MFDM
**********************************************************************
L2 Eth Receivers :
IGMP Snooping: 1/2, 1/3
MFDM: 1/2, 1/3

L2 PC Receivers :
IGMP Snooping: 100, 12, 18
MFDM: 12, 100, 18
```

```
CC between IGMP Snooping and MFDM PASSED
```

次に、IGMP スヌーピングと MFDM を比較する例（失敗）を示します。

```
*********************************************************************
Comparing IGMP Snooping with MFDM
*********************************************************************
L2 Eth Receivers:
IGMP Snooping: 1/2, 1/3
MFDM: 1/2,  1/3

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
L2 PC Receivers:
IGMP Snooping: 100, 12, 18
MFDM: 12, 100, 16
Consistency check failed!!!
Missing elements are: 18
Additional elements are: 16
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

輻輳検出および回避

Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9000 シリーズ スイッチは、輻輳の問題をトラブルシューティングするための **show tech-support slowdrain** コマンドをサポートしています。**show tech-support slowdrain** コマンドには、輻輳検出表示、カウンタ、およびログ メッセージの一部と、スイッチ、Cisco NX-OS バージョン、およびトポロジを理解できるその他のコマンドが含まれています。

輻輳は 1 つのスイッチから別のスイッチに伝播する可能性があるため、輻輳のトリガーと伝播をより適切に評価するために、すべてのスイッチから同時に **show tech-support slowdrain** コマンドの出力を収集する必要があります。

ACL 整合性チェッカ

Cisco NX-OS Release 9.3(3) 以降、ACL 整合性チェッカは次のデバイスをサポートします。

N9K-C9372PX、N9K-C9372PX-E、N9K-C9372TX、N9K-C9372TX-E、N9K-C9332PQ、N9K-C93128TX、N9K-C9396PX、N9K-C9396TX、N9K-C9508-FM-S、N9K-C9508-FM2、N9K-C9504-FM-S、N9K-X9632PC-QSFP100、N9K-X9432C-S

Cisco NX-OSリリース9.3(5) 以降、ACL 整合性チェッカは Cisco Nexus N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX、N9K-C93240YC-FX2、N9K-C93180YC-EX、N3K-C3636C-R、N3K-C36180YC-Rと、 N9K-X9636Q-R、N9K-X9636C-R、N9K-X9636C-RX および N9K-X96136YC-R ライン カードを搭載した Cisco Nexus 9500 シリーズ スイッチでサポートされています。

Cisco NX-OS リリース 10.3（1）F 以降、ACL 整合性チェッカは Cisco Nexus 9808 プラットフォーム スイッチでサポートされています。

- Cisco NX-OS リリース 10.4（1）F 以降、ACL 一貫性チェッカーは、Cisco Nexus 9808 スイッチ（Cisco Nexus X98900CD-A、X9836DM-A ライン カード搭載）サポートされます。

Cisco NX-OS リリース 10.4（1）F 以降、ACL 一貫性チェッカーは Cisco Nexus 9804 プラットフォーム スイッチ、Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

次のエンティティは、ACLの整合性チェックの一部として検証されます：

アクション、プロトコル、SIP、DIP、送信元ポート、宛先ポート、送信元MAC、宛先MAC、Ethertype、COS、DSCP、VLAN および UDF です。

Cisco NX-OS は、次の PACL、RACL、および VACL 整合性チェッカ コマンドをサポートしています。

コマンド	説明
show consistency-checker pacl extended ingress ip module <module-id> [brief | detail]	指定した IP モジュールの入力インターフェイスおよびポートチャネルの PACL 整合性チェックを実施します。
show consistency-checker pacl extended ingress ipv6 module <module-id> [brief | detail]	指定した IPv6 モジュールの入力インターフェイスおよびポートチャネルの PACL 整合性チェックを実施します。
show consistency-checker pacl extended ingress mac module <module-id> [brief | detail]	指定された MAC モジュールの入力インターフェイスおよびポートチャネルの MAC PACL 整合性チェックを実施します。
show consistency-checker pacl extended ingress ip interface {<int-id> | <ch-id> [brief | detail]	指定された 入力 インターフェイスの PACL 整合性チェックを実施します。
show consistency-checker pacl extended ingress ipv6 interface {<int-id> | <ch-id> [brief | detail]	指定された IPv6 入力インターフェイスの PACL 整合性チェックを実施します。
show consistency-checker pacl extended ingress mac interface {<int-id> | <ch-id> [brief | detail]	指定された入力 MAC インターフェイスの PACL 整合性チェックを実施します。
show consistency-checker racl extended ingress ip module <module-id> [brief | detail]	指定した IP モジュールの入力インターフェイスおよびポートチャネルの RACL 整合性チェックを実施します。
show consistency-checker racl extended ingress ipv6 module <module-id> [brief | detail]	指定された IPv6 モジュールの入力インターフェイスおよびポートチャネルの RACL 整合性チェックを実施します。
show consistency-checker racl extended ingress ip interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	指定された入力インターフェイスの RACL 整合性チェックを実施します。

コマンド	説明
show consistency-checker racl extended egress ip interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	指定された出力 IP インターフェイスの RACL 整合性チェックを実施します。
show consistency-checker racl extended ingress ipv6 interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	指定した入力 IPv6 インターフェイスの RACL 整合性チェックを実施します。
show consistency-checker racl extended egress ipv6 interface {<int-id> | <ch-id> | <vlan-id>}[brief | detail]	指定された出力 IPv6 インターフェイスの RACL 整合性チェックを実施します。
show consistency-checker vacl extended ingress ip vlan <vlan-id> [brief | detail]	指定された IP VLAN の VACL 整合性チェックを実施します。
show consistency-checker vacl extended ingress ipv6 vlan <vlan-id> [brief | detail]	指定された IPv6 VLAN の VACL 整合性チェックを実施します。
show consistency-checker vacl extended ingress mac vlan <vlan-id> [brief | detail]	指定された入力 MAC VLAN の VACL 整合性チェックを実施します。

ACL 整合性チェッカ コマンドの出力例

次に、RACL 整合性チェックの結果の例を示します。

```
switch# show consistency-checker racl extended ingress ip module 1 Consistency checker
passed for Eth1/3 (ingress, ip, ip-list)
switch#
switch#
switch# show consistency-checker racl extended ingress ip module 1 brief
{
   "result": {
   "status": "CC_STATUS_OK",
   "checkers": [
     {
       "version": 1,
       "type": "CC_TYPE_IF_RACL",
       "status": "CC_STATUS_OK",
        "platformDetails": {
        "classType": "CC_PLTFM_NXOS_BCM"
      },
      "recoveryActions": [],
      "failedEntities": []
      }
    ]
   }
}
switch#
switch # show consistency-checker racl extended ingress ip interface ethernet 3/5
Consistency checker passed for Ethernet3/5 (ingress, ip, ip-list)
switch#
switch# show consistency-checker racl extended ingress ip interface ethernet 3/5 brief
{
   "result": {
   "status": "CC_STATUS_OK",
    "checkers": [
```

```
      {
       "version": 1,
       "type": "CC_TYPE_IF_RACL",
       "status": "CC_STATUS_OK",
       "platformDetails": {
       "classType": "CC_PLTFM_NXOS_BCM"
      },
       "recoveryActions": [],
       "failedEntities": []
     }
    ]
  }
}
```


プロアクティブな整合性チェッカー

Nexusプラットフォーム上のソフトウェアテーブルとハードウェアテーブル間の整合性チェックは、ルート整合性チェッカーに関して優先度の高い保守性の課題です。既存のルート整合性チェッカーは予防的なメカニズムではなく、コマンドが発行されたときのオンデマンドの整合性チェッカーです。

プロアクティブ整合性チェッカーには、バックグラウンドで継続的に実行されるルート/隣接整合性チェッカーがあり、IPv4 または IPv6 ルートおよび ARP または ND 隣接の不整合を事前に検出できます。

Cisco NX-OS リリース 10.3（1）F 以降、プロアクティブ整合性チェッカーは R/RX カードと一緒に Cisco Nexus 9504/9508 モジュラ シャーシでサポートされています。

プロアクティブ整合性チェッカーは、すべての Cloudscale EOR および TOR プラットフォームでサポートされています。2 種類の整合性チェック方法があります。

- **フル データベース整合性チェッカー**：これは、完全なルートと隣接データベースの整合性チェックを実行します。
- **増分整合性チェッカー**：この整合性チェックは、一定期間にわたって更新または追加されたルートおよび隣接の増分変更セットに対して実行されます。

Cisco NX-OS リリース 10.3（2）F 以降、プロアクティブな整合性チェッカーは、R/R2/RX ライン カードを搭載した Cisco Nexus 9504 および 9508 モジュラ型シャーシで、IPv4、IPv6、VPNv4、VPNv6、および PE/Deagg FEC タイプの MPLS ルート整合性チェックをサポートします。

Show コマンド

プロアクティブな整合性チェッカーによって不整合が検出されるたびに、次の syslog が生成されます。

"%UFDM-3-PROACTIVE_CC_INCONSISTENCY_FOUND: プロアクティブ CC セッションで矛盾が見つかりました"

プロアクティブな整合性チェック中に不整合をチェックするには、次の 2 つのコマンドを使用する必要があります。

コマンド	説明
show forwarding proactive-cc inconsistencies	この show コマンドは、最後に失敗した反復で見つかった不整合を表示します。
show forwarding proactive-cc inconsistencies all	この show コマンドは、プロアクティブな整合性チェックが設定された時点から見つかったすべての不整合を表示します

ユーザーが上記の2つのコマンドに見られる不整合を解消したい場合は、次のコマンドを使用できます。

"clear forwarding proactive-cc inconsistencies"

コンフィギュレーション コマンド

以下は、機能を有効化/無効化し、増分および完全な整合性チェックの周期 (タイマー) を変更するコマンドです。

- **platform proactive-cc forwarding**（デフォルト タイマーで有効化）
- **no platform proactive-cc forwarding**（無効にする）
- **プラットフォームのプロアクティブ cc 転送 fulldb <time in sec>**
- **platform proactive-cc forwarding incremental <time in sec>**
- **platform proactive-cc forwarding incremental <time in sec> fulldb <time in sec>**

コマンド	目的
platform proactive-cc forwarding 例： `switch(config)# platform proactive-cc forwarding`	このコマンドにより、スイッチのプロアクティブな整合性チェッカーが有効になり、デフォルトのタイマーが設定されます。 FulldB のデフォルトのタイマー値は 86400 です。 増分 dB デフォルト タイマー値は 10 秒です。
no platform proactive-cc forwarding 例： `switch(config)# no platform proactive-cc forwarding`	このコマンドは、プロアクティブな整合性チェッカーを無効にします。

コマンド	目的
platform proactive-cc forwarding fulldb <time in sec> **例**： `switch(config)# platform proactive-cc forwarding`	このコマンドは、プロアクティブな整合性チェッカーの fulldB タイマーを 600 秒に設定します。
platform proactive-cc forwarding incremental <time in sec> **例**： `switch(config)# platform proactive-cc forwarding incremental 20`	このコマンドは、プロアクティブ cc 増分タイマー値を 20 秒に設定します。
platform proactive-cc forwarding incremental <time in sec> fulldb <time in sec> **例**： `switch(config)# platform proactive-cc forwarding incremental 20 fulldb 600`	このコマンドは、増分タイマーと fulldB タイマーの両方を一緒に設定します。

インターフェイス整合性チェッカー

Cisco NX-OS リリース 10.3（1）F 以降、インターフェイス一貫性チェッカーは Cisco Nexus 9808 プラットフォーム スイッチでサポートされています。

Cisco NX-OS リリース 10.4（1）F 以降、インターフェイス一貫性チェッカーは、Cisco Nexus 9808 スイッチ（Cisco Nexus X98900CD-A、X9836DM-A ライン カード搭載）サポートされます。

Cisco NX-OS リリース 10.4（2）F 以降、インターフェイス一貫性チェッカーは、Cisco Nexus 9232E-B1 スイッチ（Cisco Nexus X98900CD-A、X9836DM-A ライン カード搭載）サポートされます。

Cisco NX-OS リリース 10.4（1）F 以降、インターフェイス一貫性チェッカーは、Cisco Nexus 9804 プラットフォーム スイッチ、Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされています。

ITD 整合性チェッカー

ITD は、予想される機能を実現するために、依存コンポーネントの設定を内部的に生成します。これらのコンポーネントで予期しない設定を行うと、ITD の誤動作が発生します。CLI を介した ITD 整合性チェッカーは、ITD とこれらのコンポーネントの実際の設定との間に不整合が見つかった場合に表示します。

ITD 整合性チェックは stop-on-error です。つまり、サービスのプロパティ チェックが機能不全になった場合、ITDは残りのプロパティのチェックをスキップし、そのサービスの失敗を返します。

例：**show consistency-checker itd all** [**brief** | **detail**] コマンドでは、1 つのサービスの 1 つのプロパティチェックが失敗した場合、ITD は次のサービスのチェックに進みます。

Cisco NX-OS リリース 10.3（2）F 以降、次の ITD 整合性チェッカー コマンドが Cisco Nexus 9300-EX / FX / FX2 / FX3 / GX / GX2 プラットフォーム スイッチでサポートされています。

コマンド	説明
show consistency-checker itd <*service-name*> [**brief** | **detail**]	1 つのサービスの整合性チェック <*service-name*>を表示します。サービスが存在しない場合、チェックはスキップされます。
show consistency-checker itd all [**brief** | **detail**]	既存の各 ITD サービスの整合性チェックを順番に表示し、各サービスのチェックが成功または機能不全になった場合の結果を含む応答を表示します。
show consistency-checker itd ingress interface <*intf-name*> **source** <*srcIP*> **destination** <*destIP*> [**brief** | **detail**]	入力インターフェイスへの特定のフローがITDサービスによって生成されたリダイレクト ポリシーにヒットした場合に、ITD サービス整合性チェッカーが成功したか機能不全になったかを表示します。フローが ITD で生成されたポリシーにヒットしていない場合、サービス整合性チェックは合格として扱われます。

設定ファイル

構成ファイルには、Cisco NX-OS デバイス上の機能を構成するために使用される Cisco NX-OS コマンドが保存されます。Cisco NX-OS には、実行構成とスタートアップ構成の 2 種類があります。デバイスは、起動時にスタートアップ コンフィギュレーション（startup-config）を使用して、ソフトウェア機能を設定します。実行コンフィギュレーション（running-config）には、スタートアップコンフィギュレーションファイルに対して行った現在の変更が保存されます。設定を変更する前に、設定ファイルのバックアップを作成してください。コンフィギュレーション ファイルはリモート サーバにバックアップできます。コンフィギュレーション ファイルの詳細については、『*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*』を参照してください。また、設定ファイルのチェックポイントコピーを作成すれば、問題が発生した場合にロールバックすることもできます。ロールバック機能については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。

Cisco NX-OS 機能は、スタートアップコンフィギュレーションファイルに内部ロックを作成することがあります。まれに、機能により作成されたロックが削除されずに残っていることがあります。**show system internal sysmgr startup-config locks** コマンドを使用して、ロックがスター

トアップ コンフィギュレーション ファイル内に残っていないか確認してください。**system startup-config unlock** コマンドを使用し、 して、これらのロックを削除してください。

CLI デバッグ

Cisco NX-OS は、ネットワークをアクティブにトラブルシューティングするための広範なデバッグ機能セットをサポートしています。CLI を使用して、各機能のデバッグモードを有効にし、リアルタイムで更新された制御プロトコル交換のアクティビティ ログを表示できます。各ログ エントリにはタイムスタンプがあり、時間順にリストされます。CLI ロール メカニズムを使用してデバッグ機能へのアクセスを制限し、ロール単位でアクセスを分割できます。**debug** コマンドはリアルタイム情報を表示するのに対し、**show** コマンドは、履歴情報とリアルタイム情報を一覧表示するために使用します。

注意 **debug** コマンドを使用し、 できるのは、シスコのテクニカル サポート担当者の指示があった場合に限られます。一部の **debug** コマンドはネットワーク パフォーマンスに影響を与える可能性があるからです。

（注） デバッグ メッセージは、特別なログ ファイルに記録できます。ログ ファイルは、デバッグ出力をコンソールに送信するよりも安全で、処理が容易です。

?オプションを使用すると、任意の機能で使用可能なオプションを表示できます。実際のデバッグ出力に加えて、入力されたコマンドごとにログ エントリが作成されます。デバッグ出力には、ローカルデバイスと他の隣接デバイス間で発生したアクティビティのタイムスタンプ付きアカウントが記録されます。

デバッグ機能を使用して、イベント、内部メッセージ、およびプロトコルエラーを追跡できます。ただし、実稼働環境でデバッグ ユーティリティを使用する場合は注意が必要です。一部のオプションは、コンソールに大量のメッセージを出力したり、ネットワーク パフォーマンスに重大な影響を与える可能性がある CPU 集約イベントを作成したりすることで、デバイスへのアクセスを妨げる可能性があります。

（注） **debug** コマンドを入力する前に、2番目の Telnet または SSH セッションを開くことを推奨します。デバッグ セッションが現在の出力ウィンドウの妨げとなる場合は、2番目のセッションを使用して **undebug all** を入力し、 デバッグ メッセージの出力を停止します。

デバッグ フィルタ

debug-filter を使用して、不要なデバッグ情報を除外できます。 コマンドを使用する必要があります。この **debug-filter** コマンドを使用すると、関連する **debug** コマンドによって生成されるデバッグ情報を制限できます。

次に、EIGRP hello パケットのデバッグ情報をイーサネット インターフェイス 2/1 に制限する例を示します。

```
switch# debug-filter ip eigrp interface ethernet 2/1
switch# debug eigrp packets hello
```

Ping、Pong、および Traceroute

（注） ping および traceroute 機能を使用して、接続およびパスの選択に関する問題をトラブルシューティングします。これらの機能を使用して、ネットワーク パフォーマンスの問題を特定または解決しないでください。2 つのポイント間のネットワークの遅延を測定するには、pong 機能を使用します。

この項で説明している **ping** および **traceroute** コマンドは、TCP/IP ネットワーキングの問題のトラブルシューティングにもっとも役立つツールの 2 つです。ping ユーティリティは、TCP/IP インターネットワークを経由する宛先に対して、一連のエコーパケットを生成します。エコーパケットは、宛先に到達すると、再ルーティングされて送信元に戻されます。

traceroute ユーティリティも同様の方法で動作しますが、ホップバイホップ ベースで宛先までの特定のパスを決定することもできます。

pong ユーティリティは、2 つのポイント間のネットワークの遅延を測定できます。

ping の使用

ping コマンドを使用し、 コマンドを使用すると、IPv4 ルーティング ネットワーク経由で特定の宛先への接続および遅延を確認できます。

ping6 コマンドを使用し、 コマンドを使用すると、IPv6 ルーティング ネットワーク経由で特定の宛先への接続および遅延を確認できます。

ping ユーティリティを使用すると、ポートまたはエンドデバイスにショートメッセージを送信できます。IPv4 または IPv6 アドレスを指定することにより、宛先に一連のフレームが送信できます。これらのフレームは、ターゲットデバイスに到達し、タイムスタンプが付加されて、送信元にループバックされます。

（注） Ping ユーティリティを使用して、システムに設定された IP アドレスでネットワーク パフォーマンスをテストすることは推奨されません。

（注） Ping ユーティリティを使用して、 Nexus スイッチに構成された IP アドレスでネットワーク パフォーマンスをテストすることは推奨されません。スイッチの IP アドレス宛ての ICMP（Ping）トラフィックは、CoPP（コントロール プレーン ポリシング）の対象となり、ドロップされる可能性があります。

```
switch# ping 172.28.230.1 vrf management
PING 172.28.230.1 (172.28.230.1): 56 data bytes
64 bytes from 172.28.230.1: icmp_seq=0 ttl=254 time=1.095 ms
64 bytes from 172.28.230.1: icmp_seq=1 ttl=254 time=1.083 ms
64 bytes from 172.28.230.1: icmp_seq=2 ttl=254 time=1.101 ms
64 bytes from 172.28.230.1: icmp_seq=3 ttl=254 time=1.093 ms
64 bytes from 172.28.230.1: icmp_seq=4 ttl=254 time=1.237 ms

--- 172.28.230.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 1.083/1.121/1.237 ms
```

トレースルートの使用

traceroute は、次の操作のために使用します。

- データ トラフィックが経由したルートを追跡します。
- スイッチ間（ホップ単位）の遅延を計算します。

traceroute ユーティリティでは、ホップごとに使用されるパスが識別され、双方向で各ホップにタイムスタンプが付けられます。traceroute を使用すると、発信元のデバイスと送信先に最も近いデバイスの間のパスに沿ってポート接続をテストできます。

traceroute {*dest-ipv4-addr* | *hostname*} [**vrf** *vrf-name*] コマンドは IPv4 ネットワーク用に、**traceroute6** {*dest-ipv6-addr* | *hostname*} [**vrf** *vrf-name*] コマンドは IPv6 ネットワーク用に使用します。送信先に到達できない場合は、パス検出によってパスが障害ポイントまで追跡されます。

```
switch# traceroute 172.28.254.254 vrf management
traceroute to 172.28.254.254 (172.28.254.254), 30 hops max, 40 byte packets
 1  172.28.230.1 (172.28.230.1)  0.941 ms  0.676 ms  0.585 ms
 2  172.24.114.213 (172.24.114.213)  0.733 ms  0.7 ms  0.69 ms
 3  172.20.147.46 (172.20.147.46)  0.671 ms  0.619 ms  0.615 ms
 4  172.28.254.254 (172.28.254.254)  0.613 ms  0.628 ms  0.61 ms
```

実行中の traceroute を終了するには、**Ctrl-C** を押します。

次のコマンドを使用して、traceroute の送信元インターフェイスを指定できます。

コマンド	目的
traceroute {*dest-ipv4-addr* | *hostname*} [**source** {*dest-ipv4-addr* | *hostname* | *interface*}] [**vrf** *vrf-name*] **例**： `switch# traceroute 112.112.112.1 source vlan 10`	指定した IP アドレス、ホスト名、またはインターフェイスからの、traceroute パケットの送信元 IPv4 アドレスを指定します。
traceroute6 {*dest-ipv6-addr* | *hostname*} [**source** {*dest-ipv6-addr* | *hostname* | *interface*}] [**vrf** *vrf-name*] **例**： `switch# traceroute6 2010:11:22:0:1000::1 source ethernet 2/2`	指定した IP アドレス、ホスト名、またはインターフェイスからの、traceroute6 パケットの送信元 IPv6 アドレスを指定します。
[**no**] **ip traceroute source-interface** *interface* [**vrf** *vrf-name*] **例**： `switch(config)# ip traceroute source-interface loopback 1`	設定されたインターフェイスから送信元 IP アドレスを持つ traceroute または traceroute6 パケットを生成します。

show ip traceroute source-interface [**vrf** *vrf-name*]
例：
`switch# show ip traceroute source-interface vrf all`
`VRF Name Interface`
`default loopback1`
	traceroute のために設定された送信元インターフェイスを表示します。

ip icmp-errors source-interface *interface*
例 1：
`switch(config)# ip icmp-errors source-interface loopback 1`
例 2：
`switch(config)# vrf context vrf-blue`
`switch(config-vrf)# ip icmp-errors source-interface loopback 2`
	設定されたインターフェイスから送信元 IPv4 または IPv6 アドレスを持つ ICMP エラー パケットを生成します。
	また、Virtual Routing and Forwarding（VRF）インスタンス内のスタティック ルートでの BFD を設定することもできます。

プロセスおよび CPU のモニタリング

show processes コマンドを使用し、すれば、実行中のプロセスおよび各プロセスのステータスを確認できます。コマンド出力には次が含まれます。

- PID = プロセス ID
- State = プロセスの状態
- PC = 現在のプログラム カウンタ（16 進形式）
- Start_cnt = プロセスがこれまでに開始（または再開）された回数
- TTY = プロセスを制御している端末通常、「-」（ハイフン）は、特定の TTY 上で実行されていないデーモンを表します。
- Process = プロセスの名前

プロセスの状態は次のとおりです。

- D = 中断なしで休止（通常 I/O）
- R = 実行可能（実行キュー上）
- S = 休止中
- T = トレースまたは停止
- Z = 機能していない (「ゾンビ」) プロセス
- NR = 実行されていない
- ER = 実行されているべきだが、現在は実行されていない

（注） 一般に、ER 状態は、プロセスの再起動回数が多すぎるために、システムが障害発生と判断してそのプロセスをディセーブルにしたことを示しています。

```
switch# show processes ?
cpu      Show processes CPU Info
log      Show information about process logs
memory   Show processes Memory Info

switch# show processes
PID    State  PC        Start_cnt    TTY   Type  Process
-----  -----  --------  -----------  ----  ----  -------------
    1      S  b7f9e468            1     -     O  init
    2      S         0            1     -     O  migration/0
    3      S         0            1     -     O  ksoftirqd/0
    4      S         0            1     -     O  desched/0
    5      S         0            1     -     O  migration/1
    6      S         0            1     -     O  ksoftirqd/1
    7      S         0            1     -     O  desched/1
```

```
   8       S          0            1      -      O  events/0
   9       S          0            1      -      O  events/1
  10       S          0            1      -      O  khelper
  15       S          0            1      -      O  kthread
  24       S          0            1      -      O  kacpid
 103       S          0            1      -      O  kblockd/0
 104       S          0            1      -      O  kblockd/1
 117       S          0            1      -      O  khubd
 184       S          0            1      -      O  pdflush
 185       S          0            1      -      O  pdflush
 187       S          0            1      -      O  aio/0
 188       S          0            1      -      O  aio/1
 189       S          0            1      -      O  SerrLogKthread

...
```

show processes cpu コマンドの使用

show processes cpu コマンドを使用し、 コマンドを使用して、CPU 利用率を表示します。コマンド出力には次が含まれます。

- Runtime(ms) = プロセスが使用した CPU 時間（ミリ秒単位）
- Invoked = プロセスがこれまでに開始された回数
- uSecs = プロセスの呼び出しごとの平均 CPU 時間（ミリ秒単位）
- 1Sec = 最近の 1 秒間における CPU 使用率（パーセント単位）

```
switch# show processes cpu
PID    Runtime(ms)  Invoked   uSecs  1Sec   Process
-----  -----------  --------  -----  -----  -----------
    1         2264    108252     20      0  init
    2          950    211341      4      0  migration/0
    3         1154  32833341      0      0  ksoftirqd/0
    4          609    419568      1      0  desched/0
    5          758    214253      3      0  migration/1
    6         2462  155309355     0      0  ksoftirqd/1
    7         2496    392083      6      0  desched/1
    8          443    282990      1      0  events/0
    9          578    260184      2      0  events/1
   10           56      2681     21      0  khelper
   15            0        30     25      0  kthread
   24            0         2      5      0  kacpid
  103           81        89    914      0  kblockd/0
  104           56       265    213      0  kblockd/1
  117            0         5     17      0  khubd
  184            0         3      3      0  pdflush
  185         1796    104798     17      0  pdflush
  187            0         2      3      0  aio/0
  188            0         2      3      0  aio/1
  189            0         1      3      0  SerrLogKthread
...
```

show system resources コマンドの使用

show system resources コマンドを使用し、 すれば、システム関連の CPU およびメモリの統計情報を表示できます。このコマンドの出力には、次の情報が表示されます。

- 実行中プロセスの平均数として定義された負荷。Load averageには、過去1分間、5分間、および15分間のシステム負荷が表示されます。
- Processesには、システム内のプロセス数、およびコマンド発行時に実際に実行されていたプロセス数が表示されます。
- CPU statesには、直前の1秒間におけるCPUのユーザ モードとカーネル モードでの使用率およびアイドル時間がパーセントで表示されます。
- Memory usageには、合計メモリ、使用中メモリ、空きメモリ、バッファに使用されているメモリ、およびキャッシュに使用されているメモリがキロバイト単位で表示されます。また、buffersおよびcacheの値には、使用中メモリの統計情報も含まれます。

```
switch# show system resources
Load average:   1 minute: 0.00   5 minutes: 0.02   15 minutes: 0.05
Processes   :   355 total, 1 running
CPU states  :   0.0% user,   0.2% kernel,   99.8% idle
        CPU0 states  :   0.0% user,   1.0% kernel,   99.0% idle
        CPU1 states  :   0.0% user,   0.0% kernel,   100.0% idle
        CPU2 states  :   0.0% user,   0.0% kernel,   100.0% idle
        CPU3 states  :   0.0% user,   0.0% kernel,   100.0% idle
Memory usage:   16402560K total,   2664308K used,   13738252K free
Current memory status: OK
```

オンボード障害ロギングの使用

Cisco NX-OSでは、障害データを永続的ストレージに記録する機能が提供されます。この記録は、分析用に取得したり、表示したりできます。このOBFL機能は、障害および環境情報をモジュールの不揮発性メモリに保管します。この情報は、障害モジュールの分析に役立ちます。

OBFL機能によって保存されるデータは、次のとおりです。

- 初期電源オンの時間
- モジュールのシャーシ スロット番号
- モジュールの初期温度
- ファームウェア、BIOS、FPGA、およびASICのバージョン
- モジュールのシリアル番号
- クラッシュのスタック トレース
- CPU hog情報
- メモリ リーク情報
- ソフトウェア エラー メッセージ
- ハードウェア例外ログ
- 環境履歴

- OBFL 固有の履歴情報
- ASIC 割り込みおよびエラー統計の履歴
- ASIC レジスタ ダンプ

OBFL の設定の詳細については、『*Cisco Nexus 9000 Series NX-OS* システム管理設定』を参照してください。

OBFL エラー ステータス コマンドの使用

Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9000 シリーズ スイッチはさまざまなカウンタをサポートし、ファイバ チャネル インターフェイスをモニタし記録します。カウンタは、FCMAC レベルでの問題の特定とトラブルシューティングに役立ちます。

show logging onboard error-stats コマンドを使用し、 コマンドはオンボード エラー統計情報を表示します。出力には、次のカウンタが含まれます。

- FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER
- FCP_CNTR_MAC_RX_EOFA
- FCP_CNTR_MAC_RX_CRC
- FCP_CNTR_MAC_RX_MAX_FRAME_TRUNCATE
- FCP_CNTR_MAC_RX_MIN_FRAME_PAD
- FCP_CNTR_CREDIT_LOSS
- FCP_CNTR_TX_WT_AVG_B2B_ZERO

次に、この **show logging onboard error-stats** コマンドの出力例を示します。

```
switch# show logging onboard error-stats
----------------------------
    Module:  1
----------------------------


-------------------------------------------------------------------------------------------------
ERROR STATISTICS INFORMATION FOR DEVICE: FCMAC
-------------------------------------------------------------------------------------------------
Interface Range    |                                          |          |Time Stamp
                   |            Error Stat Counter Name       |  Count   |MM/DD/YY HH:MM:SS

                   |                                          |          |
-------------------------------------------------------------------------------------------------
fc1/9              |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4         |11/15/19 09:54:40
fc1/33             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4         |11/15/19 09:37:53
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4         |11/15/19 09:05:13
fc1/37             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4         |11/15/19 08:42:56
fc1/37             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4         |11/15/19 08:21:19
fc1/28             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4         |11/15/19 08:20:59
fc1/9              |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |5996      |11/14/19 10:25:45
fc1/9              |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |5992      |11/14/19 06:19:04
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |22112     |11/14/19 06:19:04
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |21876     |11/14/19 06:18:44
```

```
fc1/36            |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |21368    |11/14/19 06:18:24
fc1/36            |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |20872    |11/14/19 06:18:04
fc1/36            |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |20292    |11/14/19 06:17:44
fc1/36            |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |19720    |11/14/19 06:17:24
fc1/36            |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |19284    |11/14/19 06:17:04
fc1/36            |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |18788    |11/14/19 06:16:44
```

診断の使用

Cisco Generic Online Diagnostics（GOLD）では、複数のシスコ プラットフォームにまたがる診断操作の共通フレームワークを定義しています。GOLDの実装により、ハードウェア コンポーネントの健全性を確認し、システム データおよびコントロール プレーンの動作の適切性を検証できます。テストにはシステムの起動時に有効になるものと、システムの実行中に有効になるものがあります。ブート モジュールは、オンラインになる前に一連のチェックを実行して、システムの起動時にハードウェア コンポーネントの障害を検出し、障害のあるモジュールが稼働中のネットワークに導入されないようにします。

システムの動作時または実行時にも不具合が診断されます。一連の診断チェックを設定して、オンライン システムの状態を確認できます。中断を伴う診断テストと中断を伴わない診断テストを区別する必要があります。中断のないテストはバックグラウンドで実行され、システム データまたはコントロール プレーンには影響しませんが、中断のあるテストはライブ パケット フローに影響します。特別なメンテナンス期間中に中断テストをスケジュールする必要があります。この項で説明している **show diagnostic content module** コマンド出力には、中断を伴うテストや中断を伴わないテストなどのテスト属性が表示されます。

ランタイム診断チェックは、特定の時刻に実行するか、バックグラウンドで継続的に実行するように設定できます。

ヘルス モニタリング診断テストは中断を伴わず、システムの動作中にバックグラウンドで実行されます。オンライン診断ヘルス モニタリングの役割は、ライブ ネットワーク環境でハードウェア障害を予防的に検出し、障害を通知することです。

GOLDは、すべてのテストの診断結果と詳細な統計情報を収集します。これには、最後の実行時間、最初と最後のテスト合格時間、最初と最後のテスト失敗時間、合計実行回数、合計失敗回数、連続失敗回数、およびエラー コードが含まれます。これらのテスト結果は、管理者がシステムの状態を判断し、システム障害の原因を理解するのに役立ちます。**show diagnostic result** コマンドを使用し、 コマンドを使用して、診断結果を表示します。

GOLD の設定の詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。

組み込まれている Event Manager の使用

Embedded Event Manager（EEM）は、主要なシステム イベントをモニタし、設定されたポリシーを介してそれらのイベントを処理できるポリシーベースのフレームワークです。ポリシーは、設定されたイベントの発生に基づいてデバイスが呼び出すアクションを定義する、ロード可能な事前にプログラムされたスクリプトです。このスクリプトは、カスタム syslog または

SNMP トラップの生成、CLI コマンドの呼び出し、フェールオーバーの強制などを含むアクションを生成できます。

EEM の設定の詳細については、「*Cisco Nexus 9000* シリーズ *NX-OS* システム管理設定ガイド」を参照してください。

Ethanalyzer の使用

Ethanalyzer は、Wireshark（旧称 Ethereal）のターミナル バージョンであるオープン ソース ソフトウェア TShark の Cisco NX-OS プロトコル アナライザツール実装です。Ethanalyzer を使用して、すべてのNexusプラットフォームのインバンドおよび管理インターフェイス上のコントロールプレーン トラフィックをキャプチャおよび分析することで、ネットワークのトラブルシューティングを行うことができます。

（注） ポートチャネルにバンドルされているインターフェイスの **前面パネル** オプションを使用した Ethanalyzer の実行はサポートされていません。代わりに、**port-channel** オプションを使用してください。

Cisco NX-OS リリース 10.3(1)F 以降、Cisco Nexus 9808 プラットフォーム スイッチで Ethanalyzer のサポートが提供されます。

- Cisco NX-OS リリース 10.4（1）F 以降、Ethanalyzer は X9836DM-A ライン カードを搭載した Cisco Nexus X98900CD-A および Cisco Nexus 9808 スイッチでサポートされます。

Cisco NX-OS リリース 10.4（1）F 以降、Ethanalyzer は Cisco Nexus 9804 プラットフォーム スイッチ、Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

Ethanalyzer を設定するには、次のコマンドを使用します。

コマンド	目的
ethanalyzer local interface inband	インバンドインターフェイスを介してスーパーバイザによって送受信されたパケットをキャプチャし、キャプチャされたパケットの要約プロトコル情報を表示します。
ethanalyzer local interface inband-in	インバンドインターフェイスを介してスーパーバイザが受信したパケットをキャプチャし、キャプチャされたパケットの要約プロトコル情報を表示します。
ethanalyzer local interface inband-out	スーパーバイザからインバンドインターフェイスを介して送信されたパケットをキャプチャし、キャプチャされたパケットのプロトコル情報のサマリーを表示します。

コマンド	目的
ethanalyzer local interface mgmt	管理インターフェイスを介して送受信されたパケットをキャプチャし、キャプチャされたパケットのプロトコル情報のサマリーが表示されます。
ethanalyzer local interface front-panel	レイヤ3（ルーテッド）前面パネルポートを介してスーパーバイザによって送受信されたパケットがキャプチャされ、キャプチャされたパケットのプロトコル情報のサマリー情報が表示されます。 （注） このコマンドは、レイヤ2（スイッチポート）前面パネルポートを介してスーパーバイザが送受信するパケットのキャプチャをサポートしません。
ethanalyzer local interface port-channel	スーパーバイザがレイヤ3（ルーテッド）ポートチャネルインターフェイスを介して送受信したパケットをキャプチャし、キャプチャしたパケットのプロトコル情報のサマリーを表示します。 （注） このコマンドは、スーパーバイザがレイヤ 2（スイッチポート）ポートチャネルインターフェイスを介して送受信するパケットのキャプチャをサポートしていません。
ethanalyzer local interface vlan	スーパーバイザがレイヤ 3 スイッチ仮想インターフェイス（SVI）を介して送受信したパケットをキャプチャし、プロトコル情報のサマリーを表示します。
ethanalyzer local interface netstack	Netstack ソフトウェアコンポーネントを介してスーパーバイザによって送受信されたパケットをキャプチャし、プロトコル情報のサマリーを表示します。
{| | | | | | } **ethanalyzer local** **interface** **mgmt** **front-panel** **inband** **port-channel** **vlan** **mgmt** **netstack** **capture** **limit-frames**	Ethanalyzer セッション内でキャプチャするフレーム数を制限します。フレーム数には、0ー500,000 の整数値を指定できます。0 を指定すると、Ethanalyzer セッションが自動的に停止する前に最大 500,000 フレームがキャプチャされます。

コマンド	目的
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **limit-frame-size** *bytes*	キャプチャするフレームの長さを制限します。フレームの長さは、192～65,536 の整数値にすることができます。
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **capture-filter** *filter*	Berkeley Packet Filter（BPF）構文を使用してキャプチャするパケットのタイプをフィルタリングします。
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **display-filter** *filter*	Wireshark または TShark表示フィルタを使用して、表示するキャプチャされたパケットのタイプをフィルタリングします。
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **write** *file*	キャプチャしたデータをファイルに保存します。有効なストレージオプションには、スイッチのブート フラッシュ、ログ フラッシュ、USB ストレージデバイス、または揮発性ストレージがあります。
ethanalyzer local read	キャプチャされたデータ ファイルを開いて分析ファイルを。有効なストレージ オプションには、スイッチのブート フラッシュ、ログ フラッシュ、USB ストレージデバイス、または揮発性ストレージがあります。
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **autostop**	Ethanalyzer セッションを自動的に停止する条件を指定します。セッションの継続時間（秒）、**write** キーワードを使用してキャプチャパケットをファイルに書き込むときにキャプチャするファイル数、および **write** キーワードを使用してキャプチャパケットをファイルに書き込むときにファイルサイズを指定できます。
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **capture-ring-buffer**	Ethanalyzer のキャプチャリング バッファ オプションを指定します。このオプションは、write キーワードと組み合わせて使用すると、リング バッファ内の 1 つ以上のファイルに継続的に書き込まれます。新しいファイルに書き込む前に Ethanalyzer が待機する時間（秒単位）、リング バッファの一部として保持するファイルの数、およびリングバッファ内の個々のファイルのファイルサイズを指定できます。
{| | | | | | } **ethanalyzer local interface** {**fex-fabric** | **front-panel** | **inband** | **mgmt** | **out-of-band** | **port-channel** *number*} **detail**	キャプチャしたパケットの詳細なプロトコル情報を表示します。

コマンド	目的
{| | | | | | } **ethanalyzer local interface** **front-panel** **inband** **inband-in** **inband-out** **mgmt** **port-channel** **vlan** **raw**	キャプチャされたパケットを 16進数形式で表示します。
{| | | | | | } **ethanalyzer local interface** **front-panel** **inband** **inband-in** **inband-out** **mgmt** **port-channel** **vlan** **vrf**	レイヤ 3 インターフェイスがデフォルト以外の VRF にある場合に、レイヤ 3 インターフェイスがメンバーである VRF を指定します。

ガイドラインと制約事項

- レイヤ 3 インターフェイスがデフォルト以外の VRF のメンバーであり、Ethanalyzer セッションで指定されている場合（たとえば、**ethanalyzer local interface front-panel ethernet1/1** または **ethanalyzer local interface port-channel1** コマンドを使用）、**vrf** キーワードを使用して、レイヤ 3 インターフェイスが Ethanalyzer セッション内のメンバーである VRF を指定する必要があります。たとえば、スーパーバイザが VRF「red」のレイヤ 3 前面パネルポート Ethernet1/1 を介して受信または送信したパケットをキャプチャするには、**ethanalyzer local interface front-panel ethernet1/1 vrf red** コマンドを使用します。
- ファイルへの書き込み時に、Ethanalyzer セッションが 500,000 パケットをキャプチャした場合、またはファイルのサイズが 11 MB に達した場合、Ethanalyzer は自動的に停止します。

例

```
switch(config)# ethanalyzer local interface inband
<CR>
> Redirect it to a file
>> Redirect it to a file in append mode
autostop Capture autostop condition
capture-filter Filter on ethanalyzer capture capture-ring-buffer Capture ring buffer
option
decode-internal Include internal system header decoding detail Display detailed protocol
 information
display-filter Display filter on frames captured
limit-captured-frames Maximum number of frames to be captured (default is 10)
limit-frame-size Capture only a subset of a frame
mirror Filter mirrored packets
raw Hex/Ascii dump the packet with possibly one line summary
write Filename to save capture to
| Pipe command output to filter

switch(config)# ethanalyzer local interface inband Capturing on 'ps-inb'

1 2021-07-26 09:36:36.395756813 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 64 PRI:
7 DEI: 0 ID: 4033
2 2021-07-26 09:36:36.395874466 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 205 PRI:
7 DEI: 0 ID: 4033
4 3 2021-07-26 09:36:36.395923840 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 806 PRI:
7 DEI: 0 ID: 4033
4 2021-07-26 09:36:36.395984384 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 1307 PRI:
7 DEI: 0 ID: 4033
5 2021-07-26 09:37:36.406020552 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 64 PRI:
7 DEI: 0 ID: 4033
```

```
6 2021-07-26 09:37:36.406155603 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 205 PRI:
7 DEI: 0 ID: 4033
7 2021-07-26 09:37:36.406220547 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 806 PRI:
7 DEI: 0 ID: 4033
8 8 2021-07-26 09:37:36.406297734 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 1307
PRI: 7 DEI: 0 ID: 4033
9 2021-07-26 09:38:36.408983263 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 64 PRI:
7 DEI: 0 ID: 4033
10 10 2021-07-26 09:38:36.409101470 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 205
PRI: 7 DEI: 0 ID: 4033
```

詳細なプロトコル情報を表示するには、「**detail**オプションを使用します必要に応じて、キャプチャの途中で Ctrl + C を使用して中止し、スイッチプロンプトを戻すことができます。

```
switch(config)# ethanalyzer local interface inband detail
Capturing on 'ps-inb'
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface ps-inb,
 id 0
Interface id: 0 (ps-inb) Interface name: ps-inb
Encapsulation type: Ethernet (1)
Arrival Time: Jul 26, 2021 11:54:37.155791496 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1627300477.155791496 seconds
[Time delta from previous captured frame: 0.000000000 seconds] [Time delta from previous
 displayed frame: 0.000000000 seconds] [Time since reference or first frame: 0.000000000
 seconds] Frame Number: 1
Frame Length: 64 bytes (512 bits)
Capture Length: 64 bytes (512 bits) [Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:vlan:ethertype:data] Ethernet II, Src:
00:22:bd:cf:b9:01, Dst: 00:22:bd:cf:b9:00
Destination: 00:22:bd:cf:b9:00 Address: 00:22:bd:cf:b9:00
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast) Source:
00:22:bd:cf:b9:01
Address: 00:22:bd:cf:b9:01
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast) Type: 802.1Q Virtual
 LAN (0x8100)
802.1Q Virtual LAN, PRI: 7, DEI: 0, ID: 4033
111. .... .... .... = Priority: Network Control (7) 4 ...0 .... .... .... = DEI: Ineligible
.... 1111 1100 0001 = ID: 4033
Type: Unknown (0x3737) Data (46 bytes)

0000 a9 04 00 00 7d a2 fe 60 47 4f 4c 44 00 0b 0b 0b ....}..`GOLD....
0010 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b ................


0020 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b ..............
Data: a90400007da2fe60474f4c44000b0b0b0b0b0b0b0b0b0b0b… [Length: 46]
```

キャプチャ中に表示するか、あるいはディスクに保存するパケットを選択するには、「**capture-filter**オプションを使用します。キャプチャ フィルタは、フィルタ処理中に高率のキャプチャを維持します。パケットの完全な分析は行われていないので、フィルタ フィールドはあらかじめ決められており、限定されています。

キャプチャ ファイルのビューを変更するには、**display-filter**オプションを使用します。ディスプレイ フィルタでは、完全に分割されたパケットを使用するため、ネットワーク トレース ファイルを分析する際に非常に複雑かつ高度なフィルタリングを実行できます。Ethanalyzer は、キャプチャしたデータを他のファイルに書き込むように指示されていない場合、キャプ

チャしたデータを一時ファイルに書き込みます。この一時ファイルは、**capture-filter** オプションに一致するすべてのパケットが一時ファイルに書き込まれますが、**display-filter** オプションに一致するパケットのみが表示されるため、ユーザの知らない間に表示フィルタが使用されるとすぐにいっぱいになります。

この例では、**limit-captured-frames** が 5 に設定されています。**capture-filter** オプションを使用すると、Ethanalyzer では、フィルタ **host 10.10.10.2** に一致する 5 つのパケットを表示します。「**display-filter**オプションを使用すると、Ethanalyzer では、まず 5 つのパケットをキャプチャし、フィルタ「**ip.addr==10.10.10.2**」に一致するパケットのみを表示します。

```
switch(config)# ethanalyzer local interface inband capture-filter "host 10.10.10.2"
limit-captured-frames 5
Capturing on inband
2013-02-10 12:51:52.150404 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:52.150480 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:52.496447 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:52.497201 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:53.149831 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200
5 packets captured
switch(config)# ethanalyzer local interface inband display-filter "ip.addr==10.10.10.2"
 limit-captured-frame 5
Capturing on inband
2013-02-10 12:53:54.217462 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:53:54.217819 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination
port:
3200
2 packets captured
```

write オプションを使用して、後で分析するために Cisco Nexus 9000 シリーズ スイッチ上のストレージデバイスの 1 つ（boothflash、logflash など）にあるファイルにキャプチャ データを書き込むことができます。キャプチャ ファイルのサイズは、10 MB に制限されます。

「**write**」オプションを使用した Ethanalyzer のコマンド例は、**ethanalyzer local interface inband writebootflash:***capture_file_name* です。次は **capture-filter**を使用した **write** オプションの例と **first-capture** の出力ファイル名を示します。

```
switch(config)# ethanalyzer local interface inband capture-filter "host 10.10.10.2"
limit-captured-frame 5 write ?
bootflash: Filename logflash: Filename slot0:     Filename
usb1:      Filename
usb2: Filename volatile: Filename
switch(config)# ethanalyzer local interface inband capture-filter "host 10.10.10.2"
limit-captured-frame 5 write bootflash:first-capture
```

キャプチャデータがファイルに保存されるとき、デフォルトでは、キャプチャされたパケットはターミナル ウィンドウに表示されません。「**display** オプションを使用すると、Cisco NX-OS では、キャプチャ データをファイルに保存しながら、パケットを表示します。

capture-ring-buffer オプションを使用すると、指定した秒数、指定したファイル数、または指定したファイルのサイズの後に複数のファイルが作成されます次に、これらのオプションの定義を示します。

```
switch(config)# ethanalyzer local interface inband capture-ring-buffer ?
duration Stop writing to the file or switch to the next file after value seconds have
elapsed
files Stop writing to capture files after value number of files were written or begin
again with the first file after value number of files were
written (form a ring buffer)
filesize Stop writing to a capture file or switch to the next file after it reaches a
size of value kilobytes
```

read オプションを使用すると、デバイス自体に保存されたファイルを読み取ることができます。

```
switch(config)# ethanalyzer local read bootflash:first-capture
2013-02-10 12:51:52.150404 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:52.150480 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:52.496447 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:52.497201 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200
2013-02-10 12:51:53.149831 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination
port:
3200

switch(config)# ethanalyzer local read bootflash:first-capture detail Frame 1 (110 bytes
 on wire, 78 bytes captured)
------------------------------SNIP-------------------------------------------------
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:data]
Ethernet II Src: 00:24:98:6f:ba:c4 (00:24:98:6f:ba:c4), Dst: 00:26:51:ce:0f:44
(00:26:51:ce:0f:44)
Destination: 00:26:51:ce:0f:44 (00:26:51:ce:0f:44) Address: 00:26:51:ce:0f:44
(00:26:51:ce:0f:44)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default) Source:
 00:24:98:ce:6f:ba:c4 (00:24:98:6f:ba:c4)
Address: 00:24:98:6f:ba:c4 (00:24:98:6f:ba:c4)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default) Type:
 IP (0x0800)
Internet Protocol, Src: 10.10.10.1 (10.10.10.1), Dst: 10.10.10.2 (10.10.10.2)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0xc0 (DSC) 0x30: Class Selector 6; ECN: 0x00)
------------------------------SNIP-------------------------------------------------
```

サーバまたは PC にファイルを転送し、ファイル。cap ファイルまたは 。pcap ファイルを読み取ることができる Wireshark や他のアプリケーションでそのファイル形式を読み取ることもできます。

```
switch(config)# copy bootflash:first-capture tftp:
Enter vrf (If no input, current vrf 'default' is considered): management
Enter hostname for the tftp server: 192.168.21.22
Trying to connect to tftp server......
```

```
Connection to Server Established. TFTP put operation was successful
Copy complete.
```

decode-internal オプションは、Nexus 9000 のパケット転送方法に関する内部情報を報告します。この情報は、CPUを通過するパケットのフローを理解し、トラブルシューティングするのに役立ちます。

```
switch(config)# ethanalyzer local interface inband decode-internal capture-filter "host
 10.10.10.2" limit-captured-frame 5 detail
Capturing on inband NXOS Protocol
NXOS VLAN: 0==================->VLAN in decimal=0=L3 interface
NXOS SOURCE INDEX: 1024 ==================->PIXN LTL source index in decimal=400=SUP
inband
NXOS DEST INDEX: 2569==================-> PIXN LTL destination index in
decimal=0xa09=e1/25 Frame 1: (70 bytes on wire, 70 bytes captured)
Arrival Time: Feb 10, 2013 22:40:02.216492000
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1627300477.155791496 seconds
[Time delta from previous captured frame: 0.000000000 seconds] [Time delta from previous
 displayed frame: 0.000000000 seconds] [Time since reference or first frame: 0.000000000
 seconds] Frame Number: 1
Frame Length: 70 bytes Capture Length: 70 bytes [Frame is marked: False]
[Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:26:51:ce:0f:43 (00:26:51:ce:0f:43), Dst: 00:24:98:6f:ba:c3
(00:24:98:6f:ba:c3)
Destination: 00:24:98:6f:ba:c3 (00:24:98:6f:ba:c3) Address: 00:24:98:6f:ba:c3
(00:24:98:6f:ba:c3)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default) Source:
 00:26:51:ce:0f:43 (00:26:51:ce:0f:43)
--------------------------------SNIP--------------------------------------------------
```

NX-OS インデックスを 16 進数に変換してから、Local Target Logic（LTL）インデックスを物理または論理インターフェイスにマップするために **show system internal pixm info ltl {index}** コマンドを使用します。

1 つの IP ホストとの間でやり取りされるトラフィックのキャプチャ

```
host 1.1.1.1
```

IP アドレスの範囲との間でやり取りされるトラフィックのキャプチャ

```
net 172.16.7.0/24
```

net 172.16.7.0 mask 255.255.255.0

IP アドレスの範囲からのトラフィックのキャプチャ

```
src net 172.16.7.0/24
```

srcnet 172.16.7.0 mask 255.255.255.0

IP アドレスの範囲へのトラフィックのキャプチャ

```
dst net 172.16.7.0/24
```

dst net 172.16.7.0 mask 255.255.255.0

UDLD、VTP、CDP のトラフィックのキャプチャ

UDLD は 単方向リンク検出、VTP は VLAN Trunking Protocol、CDP は Cisco Discovery Protocol です。

```
ether host 01□00□0c□cc□cc□cc
```

MAC アドレスとの間でやり取りされるトラフィックのキャプチャ

```
ether host 00□01□02□03□04□05
```

（注） and = &&

or = ||

Not = !

MAC address format : xx:xx:xx:xx:xx:xx

一般的なコントロール プレーン プロトコル

- UDLD: Destination Media Access Controller (DMAC) = 01-00-0C-CC-CC-CC and EthType = 0x0111
- LACP: DMAC = 01:80:C2:00:00:02 and EthType = 0x8809. LACP stands for Link Aggregation Control Protocol
- STP: DMAC = 01:80:C2:00:00:00 and EthType = 0x4242 - or - DMAC = 01:00:0C:CC:CC:CD and EthType = 0x010B
- CDP: DMAC = 01-00-0C-CC-CC-CC and EthType = 0x2000
- LLDP: DMAC = 01:80:C2:00:00:0E or 01:80:C2:00:00:03 or 01:80:C2:00:00:00 and EthType = 0x88CC
- DOT1X: DMAC = 01:80:C2:00:00:03 and EthType = 0x888E. DOT1X stands for IEEE 802.1x
- IPv6: EthType = 0x86DD
- UDP と TCP のポート番号のリスト

Ethanalyzer は、Cisco NX-OS がハードウェアで転送するデータ トラフィックはキャプチャしません。

Ethanalyzer は、**tcpdump** と同じキャプチャ フィルタ構文を使用します。 および Wireshark表示フィルタ構文を使用します。

次の例では、キャプチャされたデータ（4 パケットに限定された）を管理インターフェイス上に表示します。

```
switch(config)# ethanalyzer local interface mgmt limit-captured-frames 4
Capturing on eth1

2013-05-18 13:21:21.841182 172.28.230.2 -> 224.0.0.2 BGP Hello (state Standy)
2013-05-18 13:21:21.842190 10.86.249.17 -> 172.28.231.193 TCP 4261 > telnet [AC] Seq=0
```

```
Ack=0 Win=64475 Len=0
2013-05-18 13:21:21.843039 172.28.231.193 -> 10.86.249.17 TELNET Telnet Data ..
2013-05-18 13:21:21.850463 00:13:5f:1c:ee:80 -> ab:00:00:02:00:00 0x6002 DEC DN

Remote Console
4 packets captured
```

次の例では、1 つの HSRP パケットについてキャプチャしたデータの詳細を表示します。

```
switch(config)# ethanalyzer local interface mgmt capture-filter "udp port 1985"
limit-captured-frames 1
Capturing on eth1
Frame 1 (62 bytes on wire, 62 bytes captured)
Arrival Time: May 18, 2013 13:29:19.961280000
[Time delta from previous captured frame: 1203341359.961280000 seconds]
[Time delta from previous displayed frame: 1203341359.961280000 seconds]
[Time since reference or first frame: 1203341359.961280000 seconds]
Frame Number: 1
Frame Length: 62 bytes
Capture Length: 62 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:hsrp]

Ethernet II, Src: 00:00:0c:07:ac:01 (00:00:0c:07:ac:01), Dst: 01:00:5e:00:00:02
(01:00:5e:00:00:02)
Destination: 01:00:5e:00:00:02 (01:00:5e:00:00:02)
Address: 01:00:5e:00:00:02 (01:00:5e:00:00:02)
.... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
Source: 00:00:0c:07:ac:01 (00:00:0c:07:ac:01)
Address: 00:00:0c:07:ac:01 (00:00:0c:07:ac:01)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Type: IP (0x0800)
Internet Protocol, Src: 172.28.230.3 (172.28.230.3), Dst: 224.0.0.2 (224.0.0.2)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
1100 00.. = Differentiated Services Codepoint: Class Selector 6 (0x30)
.... ..0. = ECN-Capable Transport (ECT): 0
.... ...0 = ECN-CE: 0

Total Length: 48
Identification: 0x0000 (0)
Flags: 0x00
0... = Reserved bit: Not set
.0.. = Don't fragment: Not set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 1
Protocol: UDP (0x11)
Header checksum: 0x46db [correct]
[Good: True]
[Bad : False]

Source: 172.28.230.3 (172.28.230.3)
Destination: 224.0.0.2 (224.0.0.2)
User Datagram Protocol, Src Port: 1985 (1985), Dst Port: 1985 (1985)
Source port: 1985 (1985)
Destination port: 1985 (1985)
Length: 28
```

```
Checksum: 0x8ab9 [correct]
[Good Checksum: True]
[Bad Checksum: False]

Cisco Hot Standby Router Protocol
Version: 0
Op Code: Hello (0)
State: Active (16)
Hellotime: Default (3)
Holdtime: Default (10)
Priority: 105
Group: 1
Reserved: 0Authentication Data: Default (cisco)
Virtual IP Address: 172.28.230.1 (172.28.230.1)

1 packets captured
```

次の例では、表示フィルタを使用して、アクティブなHSRP状態のHSRPパケットのみを表示します。

```
switch(config)# ethanalyzer local interface mgmt display-filter "hsrp.state==Active"
limit-captured-frames 2
Capturing on eth1

2013-05-18 14:35:41.443118 172.28.230.3 -> 224.0.0.2 HSRP Hello (state Active)
2013-05-18 14:35:44.326892 172.28.230.3 -> 224.0.0.2 HSRP Hello (state Active)
2 packets captured
```

Ethanalyzer バックグラウンド キャプチャ プロセスおよびインバンド パケットの自動収集

Ethanalyzer は、インバンド パケットをキャプチャするバックグラウンド タスクとして実行できます。インバンド パケット データは PCAP ファイルの RAM メモリに保持されます。設定可能な制限された量のPCAPデータ（設定可能なファイル サイズで設定可能な数のファイル）をいつでも使用できます。制限に達すると、最も古いファイルが周期的に現在のキャプチャで上書きされます。

Ethanalyzer のバックグラウンドタスクによってキャプチャされたデータは RAM 内にあり、ブートフラッシュ領域を占有せずに周期的に上書きされます。ユーザがデータを確認できるようにするには、スナップショットを取得する必要があります。 RAM から表示のための不揮発性ストレージ（ブートフラッシュ）へのPCAP形式のバックグラウンドプロセスにより取得されるパケット キャプチャ情報をコピーします。スナップショットを作成する場合は、使用可能なブートフラッシュ領域を考慮する必要があります。

スナップショットは、CLIを介してユーザが手動でトリガーできます。EEMポリシーは、特定のイベントでスナップショットをトリガーするためにも使用できます。トリガーの使用例として、インバンド レートが定義されたしきい値を超えた場合、CoPP ドロップがしきい値を超えた場合などがあります。スナップショットは、イベントの発生時点までにどのパケットがインバンドにヒットしていたかを示します。

レートをモニタする場合、ユーザが通常予想するレートまたは許容レートを超えるしきい値を設定する必要があります。これは、問題以外のアラートの超過を回避するために設定する必要があります。以下の自動収集EEMポリシーで最大トリガーを増やす場合は、注意が必要です。

これらのプラクティスに従わないと、無関係な PCAP データが大量にスナップショット化され、ブートフラッシュがいっぱいになる可能性があります。

Ethanalyzer は、バックグラウンド セッションの有効化と設定、セッションの開始と停止、Ethanalyzer 情報のスナップショット、およびバックグラウンド セッション ステータスを確認するための show コマンドを追加するための CLI を追加しました。すべての CLI は有効から実行します。

Ethanalyzer バックグラウンド キャプチャ プロセスに関する注意事項と制限事項

- Ethanalyzer バックグラウンド プロセスでは、ストレージ容量が制限されている /tmp ディレクトリに .pcap ファイルを保存します。すべての `.pcap` ファイルの合計サイズが、使用可能な `/tmp` の ストレージ容量を超えないようにする必要があります。

  Ethanalyzer `.pcap` ファイルに必要な合計スペースを計算するには、次の式を使用します。

  fileSize * numFiles < Available /tmp Space

  Ethanalyzer バックグラウンド プロセスを開始する前に、次のコマンドを使用して /tmp ストレージの可用性を検証します：

  ```
  bash-4.4# df -k /tmp
  Filesystem     1K-blocks  Used Available Use% Mounted on
  none              614400  2760    611640   1% /var/volatile/tmp
  ```

- Ethanalyzer のバックグラウンド セッションを再起動すると、/tmp 内の以前にキャプチャされたすべての .pcap ファイルが削除されます。ユーザは、再起動する前に ethanalyzer copy-background-snapshot コマンドを使用して、重要なデータを永続ストレージ（/bootflash など）にコピーする必要があります。

  次のコマンドを使用して、再起動する前に .pcap ファイルをブートフラッシュにコピーします：

  ```
  ethanalyzer copy-background-snapshot
  ```

- スナップショットは /tmp からブートフラッシュにコピーされるため、スナップショットを取得する前に使用可能なブートフラッシュ領域を考慮してください。ブートフラッシュ領域が不足していると、スナップショットが失敗したり、データ ストレージが不完全になったりする可能性があります。

  スペースを節約するには、圧縮 tar オプションを使用します：

  ```
  ethanalyzer copy-compressed-background-snapshot
  ```

- Event Manager（EEM）ポリシーを使用して、インバンド レートしきい値やCoPP ドロップなどのイベントに基づいてスナップショットをトリガーします。過度なアラートや無関係なデータ スナップショットにより、ブートフラッシュが不必要にいっぱいになるのを避けるために、max-triggers パラメータは慎重に構成してください。

- バックグラウンド プロセスでは、/tmp を超えるストレージ ロールオーバーは自動的に管理されません。/tmp ストレージがいっぱいにならないようにするには、パラメータを適切に構成する必要があります。

- /tmp ストレージがいっぱいになると、収集されたデータが失われる可能性があります。

表 *2 : Ethanalyzer CLI*

CLI	説明
ethanalyzer background-session config <filename|filesize|numfiles|session>	循環バッファのキャプチャ パケットの Ethanalyzer バックグラウンド プロセス/セッションのパラメータを設定します。 • Filename: Ethanalyzer バックグラウンド キャプチャ プロセスによって保存されたバックグラウンド パケット キャプチャ ファイル名。 • Filesize: 一時バッファ内の個々のキャプチャ ファイルのサイズ。値の範囲は 1ー65536 KB です。 • Numfiles: 一時バッファに保存される最大 pcap ファイルの数。値の範囲は 2ー16 です。 • Session: Ethanalyzer バックグラウンド キャプチャ セッションを有効または無効にします。
ethanalyzer background-session restart	Ethanalyzer バックグラウンド キャプチャ セッションを開始/再起動します。
ethanalyzer background-session stop	Ethanalyzer バックグラウンド キャプチャ セッションを停止します。
show ethanalyzer background-session processes	Ethanalyzer バックグラウンド キャプチャ セッションの詳細を表示します。
show ethanalyzer background-session config	Ethanalyzer バックグラウンド キャプチャ セッション設定ファイルを出力します。
ethanalyzer copy-background-snapshot	一時バッファにキャプチャされたファイルをブートフラッシュにコピーします。ファイルは pcap 形式です。
ethanalyzer copy-compressed-background-snapshot	一時バッファにキャプチャされたファイルを tar し、tar ファイルをブートフラッシュにコピーします。 （注） この CLI を複数回発行すると、古い tar ファイルが削除されます。古い tar ファイルがブートフラッシュに存在する場合は、コピーすることを推奨します。

Cisco NX-OS リリース 10.1(2) Ethanalyzer Autocollection CLI は、すべての Cisco Nexus 9000 シリーズ プラットフォームでサポートされます。

Ethanalyzer Autocollection CLI 警告

Ethnalyzer Autocollection CLI の警告は次のとおりです。

- バックグラウンド プロセスに変更が加えられるたびに、Ethanalyzer バックグラウンド プロセスを再起動/開始する必要があります。設定が変更されると、次の警告メッセージがユーザに表示されます。

  「設定の変更を有効にするには、Ethanalyzerバックグラウンドプロセスを再起動してください。（Please restart the Ethanalyzer background process for any config change to take effect.）」

- スーパーバイザの冗長性がサポートされているプラットフォームでは、アクティブなスーパーバイザのスイッチオーバーによって、Ethanalyzerのバックグラウンドキャプチャプロセスが自動的に開始されないことがあります。ユーザは、Ethanalyzerバックグラウンドプロセスを手動で再起動する必要があります。スイッチオーバー後にEthanalyzerバックグラウンド プロセスを自動的に開始する場合は、アクティブ スーパーバイザでセッション イネーブルを設定し、スイッチをリロードして有効にする必要があります。この後、スイッチオーバーが発生した場合でも、新しくアクティブになったスーパーバイザでEthanalyzer バックグラウンド キャプチャ プロセスが自動的に開始されます。

CLI の例

CLI 出力の例：すべてのコマンドはイネーブル モードから実行されます。

ステップ 1：バックグラウンドで実行されている Ethanalyzer セッションを有効にします。

```
switch# ethanalyzer background-session config session enable
```

```
switch# dir bootflash: | include dump
       1087    Jan 29 13:55:46 2021  dumpcap_bg_session_configuration.xml
switch# show ethanalyzer background-session config
<?xml version="1.0"?>
<!-- This document contains configuration settings for background packet -->
<!-- capture session to execute in ring buffer mode. Please modify the settings
based on system resources -->
<!-- path:         background packet capture directory where ring buffer files w
ill be saved -->
<!-- filename:     background packet capture file name saved by dumpcap. Files w
ill be generated as filename_number_date format -->
<!-- filesize:     Size of individual ring buffer file in kB. Note that the file
size is limited to a maximum value of 65536 kB-->
<!-- num_of_files: value begin again with the first file after value number of f
iles were written (form a ring buffer). The maximum value should be equal to 16
-->
<!-- session:      Enable/disable background packet capture session process. App
licable for both boot-up as well as session restart -->
<ethanalyzer_config>
    <filepath>/tmp/dumpcap_bg_session_files/</filepath>
    <filename>capture</filename>
    <filesize>2048</filesize>
    <numfiles>2</numfiles>
    <session>enable</session>
</ethanalyzer_config>
```

次に、CLI の出力を示します。

```
switch# ethanalyzer background-session restart
root     30038     1  0 13:58 ttyS0    00:00:00 /usr/bin/dumpcap -n -b filesize:
2048 -b files:2 -i ps-inb -Z none -w /tmp/dumpcap_bg_session_files/capture.pcap
```

ステップ 2：バックグラウンド セッション設定パラメータの確認

```
switch# show ethanalyzer background-session process
```

ステップ 3：バックグラウンド Ethanalyzer プロセスの開始

```
switch# ethanalyzer background-session restart
```

ステップ 4：Ethanalyzer バックグラウンド キャプチャ セッションの実行の確認

```
switch# ethanalyzer background-session processes
Background session of packet analyzer:
root 17216 1 4 12:43 ttyS0 00:00:00 /usr/bin/dumpcap -n -b filesize:2048 -b files:2 -i
ps-inb -Z none -w /tmp/dumpcap_bg_session_files/capture.pcap
switch#
```

使用例：CLI を実行してスナップショットをキャプチャして表示する

```
switch# ethanalyzer copy-background-snapshot

Copy packet analyzer captured frames to bootflash...
Copied snapshot files :
     72 -rw-rw-rw-  1 root  root               65844 Jan 21 00:21
CAPTURE_00001_20210121001903.pcap

switch# ethanalyzer copy-compressed-background-snapshot

Copy packet analyzer captured compressed frames to bootflash...
Copied snapshot files :
     28 -rw-r--r--  1 root  root               27181 Jan 21 00:22 CAPTURE.tar.gz
```

使用例：Ethanalyzer スナップショットの自動収集のトリガーとしてインバンド レート モニタリングを使用する。

表 *3:* インバンド レート モニタリング *CLI* オプション

CLI	説明
設定モード	system inband cpu-mac log threshold rx rx_pps tx tx_pps throttle secondsrx_pps, tx_pps: 0-1500000 Inband rx/tx pps rate that needs to be logged when exceededseconds: log throttle interval (maximum 1 exceed log per defined interval)
有効モード（Enable Mode）	show system inband cpu-mac log threshold" to display settings
デフォルト	off（PPS 値 0）、スロットル間隔 120 秒。

前のセクションで説明したように、Ethanalyzer バックグラウンド プロセス機能が設定され、実行されていることが前提となります。この使用例にはデモまたはサンプル目的のサンプル レートがありますが、ユーザはロギングに値すると考えられる現実的なレートを使用する必要

があります。ユーザの要件を超えるしきい値は、非問題のアラートの超過を回避するために通知する必要があります。

（注） 以下の自動収集 EEM ポリシーで最大トリガーを増やす場合は注意が必要です。これらの方法に従わないと、大量の PCAP データがスナップショット化され、ブートフラッシュがいっぱいになる可能性があります。

max-triggers パラメータは、アクティブなスーパーバイザのブートフラッシュ（bootflash:eem_snapshots）の eem_snapshots ディレクトリに永続的に保存されているスナップショットファイルの量に対してチェックされます。スーパーバイザスイッチオーバーの場合、新しくアクティブになったスーパーバイザの収集数は、以前にアクティブだったスーパーバイザの収集数とは異なる場合があり、その結果、自動収集が再開されるかどうかが決まります。自動収集の再開は、新しくアクティブになったスーパーバイザのブートフラッシュに存在するスナップショット バンドルによって異なります。

指定されたディレクトリ内のファイルの量が max-triggers と一致すると、自動収集は停止します。再度開始するには、ユーザがディレクトリからスナップショットファイルを削除して、ファイル数を max-triggers よりも少ない「値」にし、別の量（max-triggers から「value」を引いた数）の自動収集を許可する必要があります。詳細については、「トリガーベースのイベントログの自動収集」の項を「Embedded Event Manager の設定」の章で参照してください。

ステップ 1：インバンド レート モニタリングを有効にする

```
switch(config)# system inband cpu-mac log threshold rx 400 tx 4000 throttle 60
switch# show system inband cpu-mac log threshold
Thresholds Rx: 400 PPS, Tx; 4000 PPS
Log throttle interval: 60 seconds
```

「トリガーベースのイベントログの自動収集」の項を「Embedded Event Manager の設定」の章で説明されているように、トリガーベースのログファイルの自動収集を利用して、ディレクトリを作成します（次の例では、ディレクトリの名前は「auto_collect」です）。 EEM ポリシーを作成または有効にすると、イベント ログと ethanalyzer pcap の組み込みスナップショット収集が有効になります。

ステップ 2：ディレクトリを作成する

```
create auto_collect directory
switch# pwd
bootflash:
switch# cd scripts
switch# mkdir auto_collect
```

ステップ 3：イベント マネージャ ポリシーを有効にする

```
switch(config)# event manager applet syslog_trigger override __syslog_trigger_default
switch(config-applet)# action 1.0 collect auto_collect rate-limit 60 max-triggers 3
$_syslog_msg
```

これにより、60 秒あたり最大 1x の自動収集が有効になり、同じトリガーに対して合計で最大 3 回、同じ syslog トリガーに対して最大 max-triggers x num_files pcap ファイルを保存します（例：3 x 2 = 6 ファイル）。

上記の使用例：大量の ICMP 要求を起動するホスト 20.1.1.100 の誤動作を特定します。

```
switch#
2021 Jan 29 15:15:27 switch %KERN-1-SYSTEM_MSG: [17181.984601] Inband Rx threshold 400
PPS reached. - kernel
2021 Jan 29 15:15:28 switch %KERN-1-SYSTEM_MSG: [17182.997911] Inband Rx threshold 400
PPS reached. - kernel
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog
Status/Secs/Logsize(Bytes)
2021-Jan-29 15:15:30  620969861    KERN-1-SYSTEM_MSG
PROCESSED:1:7118865
2021-Jan-29 15:15:30  201962781    KERN-1-SYSTEM_MSG
DROPPED-LASTACTIONINPROG
2021-Jan-29 15:15:29  620969861    KERN-1-SYSTEM_MSG                          PROCESSING
...
switch# dir bootflash: | include capture
    2048040    Jan 29 15:15:29 2021  capture_00004_20210129150732.pcap
     169288    Jan 29 15:15:29 2021  capture_00005_20210129151528.pcap
 ...
```

バックグラウンドプロセスでキャプチャされたファイルをデコードするには、シスコ TAC チームにお問い合わせください。

使用例：カスタム（非組み込みの自動コレクション YAML）トリガーの使用（CoPP ドロップしきい値超過）

前提条件は次のとおりです。

1. 前述のように、Ethanalyzer バックグラウンドプロセス機能が設定され、実行されています。

2. 前の使用例のステップ 2 とステップ 3 が完了しています。

ドロップが発生する理由を学習するクラスの CoPP しきい値ロギングを有効にします。詳細については、CoPP設定ガイド（参照）を参照してください。

この例では、ARP を含むクラス copp-class-normal の場合、しきい値は 1000000 に設定され、ロギング レベルは 1（autocollect に対応できる十分な高さ）に設定されます。

```
class copp-class-normal
    logging drop threshold 1000000 level 1
```

前の使用例で使用したものと同じディレクトリ（bootflash:scripts/auto_collect）で、ファイル copp.yaml を次のように追加します（copp = コンポーネント名）。

```
#********************************************************************
#
# File:   comp specific yaml
# Author:
#
# Description: Module Makefile
#
#
# Copyright (c) 2019 by cisco Systems, Inc.
# All rights reserved.
#
#
# $Id: comp specific yaml $
# $Source:  $
# $Author:  $
#
#********************************************************************
```

```
version: 1
components:
    copp:
            default:
            copp_drops1:
              serviceCOPP:
                match: CoPP drops exceed threshold
                commands: ethanalyzer copy-background-snapshot
```

上記の使用例：クラスで CoPP ドロップを引き起こす大量の ARP 要求を特定します。

```
switch#
2021 Jan 29 15:49:47 switch %COPP-1-COPP_DROPS1: CoPP drops exceed threshold in class:
copp-class-normal-log,
check show policy-map interface control-plane for more info.
switch# show policy-map interface control-plane class copp-class-normal-log
Control Plane

  Service-policy  input: copp-policy-strict-log

    class-map copp-class-normal-log (match-any)
      match access-group name copp-acl-mac-dot1x-log
      match protocol arp
      set cos 1
      threshold: 1000000, level: 1
      police cir 1400 kbps , bc 32000 bytes
      module 1 :
        transmitted 25690204 bytes;
        5-minute offered rate 168761 bytes/sec
        conformed 194394 peak-rate bytes/sec
          at Fri Jan 29 15:49:56 2021

        dropped 92058020 bytes;
        5-min violate rate 615169 byte/sec
        violated 698977 peak-rate byte/sec        at Fri Jan 29 15:49:56 2021
```

```
switch#
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog
Status/Secs/Logsize(Bytes)
2021-Jan-29 15:49:57  1232244872   COPP-1-COPP_DROPS1                        RATELIMITED
2021-Jan-29 15:49:50  522271686    COPP-1-COPP_DROPS1
PROCESSED:1:11182862
2021-Jan-29 15:49:48  522271686    COPP-1-COPP_DROPS1                        PROCESSING
...
switch# dir bootflash: | include capture
    2048192    Jan 29 15:49:49 2021  capture_00038_20210129154942.pcap
    1788016    Jan 29 15:49:49 2021  capture_00039_20210129154946.pcap
....
```

SSO の動作

スタンバイ スーパーバイザ がバックグラウンドプロセス設定 session = disable で起動した場合、ユーザはこの スーパーバイザ がアクティブになったときにプロセスを再起動する必要があります。

参考資料

- Wireshark：CaptureFilters
- Wireshark：DisplayFilters

- 『Cisco Nexus 9000 シリーズ NX-OS Layer 2 スイッチング設定ガイド』
- 『Cisco Nexus 9000 シリーズ NX-OS VXLAN 設定ガイド』
- 『Cisco Nexus 9000 NX-OS インターフェイス設定ガイド』
- 『Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング設定ガイド』

SNMP および RMON のサポート

Cisco NX-OS は、管理情報ベース（MIB）と通知（トラップと情報）を含む広範な SNMPv1、v2、および v3 のサポートを提供します。

SNMP 標準では、Cisco NX-OS を管理しモニタリングする各 MIB をサポートするサードパーティ製アプリケーションを使用できます。

SNMPv3 はさらに広範なセキュリティ機能を提供します。各デバイスで SNMP サービスを有効または無効にするように選択できます。また、各デバイスで SNMP v1 および v2 要求の処理方法を設定できます。

Cisco NX-OS は、リモート モニタリング（RMON）アラームおよびイベントもサポートします。RMON アラームとイベントは、ネットワーク動作の変化に基づいて、しきい値の設定や通知の送信のメカニズムを提供します。

[アラーム グループ（Alarm Group）] では、アラームを設定できます。アラームは、デバイス内の 1 つまたは複数のパラメータに設定できます。たとえば、デバイスの CPU 使用率の特定のレベルに対して RMON アラームを設定できます。*EventGroup* を使用すると、アラーム条件に基づいて実行するアクションであるイベントを設定できます。サポートされるイベントのタイプには、ロギング、SNMP トラップ、およびログアンドトラップが含まれます。

SNMP および RMON の設定の詳細については、「*Cisco Nexus 9000* シリーズ *NX-OS* システム管理設定ガイド」を参照してください。

PCAP SNMP パーサーの使用

PCAP SNMP パーサーは、.pcap 形式でキャプチャされた SNMP パケットを分析するツールです。スイッチ上で動作し、スイッチに送信されるすべての SNMP get、getnext、getbulk、set、trap、および response 要求の統計情報レポートを生成します。

PCAP SNMP パーサーを使用するには、次のいずれかのコマンドを使用します。

- **debug packet-analysis snmp** [**mgmt0** | **inband**] **duration** *seconds* [*output-file*] [**keep-pcap**]—Tshark を使用して指定の秒数間のパケットをキャプチャし、一時 .pcap ファイルに保存します。次に、その .pcap ファイルに基づいてパケットを分析します。

  結果は出力ファイルに保存されます。出力ファイルが指定されていない場合は、コンソールに出力されます。**keep-pcap** オプションを使用する場合を除き、一時 .pcap ファイルはデ

フォルトで削除されます。パケット キャプチャは、デフォルトの管理インターフェイス（mgmt0）、または帯域内インターフェイスで実行できます。

例：

```
switch# debug packet-analysis snmp duration 100

switch# debug packet-analysis snmp duration 100 bootflash:snmp_stats.log

switch# debug packet-analysis snmp duration 100 bootflash:snmp_stats.log keep-pcap

switch# debug packet-analysis snmp inband duration 100

switch# debug packet-analysis snmp inband duration 100 bootflash:snmp_stats.log

switch# debug packet-analysis snmp inband duration 100 bootflash:snmp_stats.log
keep-pcap
```

- **debug packet-analysis snmp** *input-pcap-file* [*output-file*]：既存の .pcap ファイルにあるキャプチャしたパケットを分析します。

  例：

```
switch# debug packet-analysis snmp bootflash:snmp.pcap

switch# debug packet-analysis snmp bootflash:snmp.pcap bootflash:snmp_stats.log
```

次に、**debug packet-analysis snmp** [**mgmt0** | **inband**] **duration** コマンドの統計情報レポートの例を示します。：

```
switch# debug packet-analysis snmp duration 10
Capturing on eth0
36
wireshark-cisco-mtc-dissector: ethertype=0xde09, devicetype=0x0
wireshark-broadcom-rcpu-dissector: ethertype=0xde08, devicetype=0x0

Started analyzing. It may take several minutes, please wait!

Statistics Report
----------------------------------------
SNMP Packet Capture Duration: 0 seconds
Total Hosts: 1
Total Requests: 18
Total Responses: 18
Total GET: 0
Total GETNEXT: 0
Total WALK: 1 (NEXT: 18)
Total GETBULK: 0
Total BULKWALK: 0 (BULK: 0)
Total SET: 0
Total TRAP: 0
Total INFORM: 0

Hosts        GET  GETNEXT  WALK(NEXT) GETBULK  BULKWALK(BULK) SET  TRAP  INFORM  RESPONSE
-------------------------------------------------------------------------------------------
10.22.27.244   0     0        1(18)      0         0(0)         0    0      0       18

Sessions
--------
1
```

```
MIB Objects GET  GETNEXT  WALK(NEXT) GETBULK(Non_rep/Max_rep) BULKWALK(BULK,
Non_rep/Max_rep)
-------------------------------------------------------------------------------------------
ifName      0       0        1(18)      0                        0


SET     Hosts
--------------------
0       10.22.27.244
```

RADIUS を利用

RADIUS プロトコルは、ヘッドエンドの RADIUS サーバとクライアント デバイス間で、属性またはクレデンシャルを交換するために使用されるプロトコルです。これらの属性は、次の 3 つのサービス クラス（CoS）に関連しています。

- 認証
- 許可
- アカウンティング

認証は、特定のデバイスにアクセスするユーザの認証を意味しています。RADIUS を使用して、Cisco NX-OS デバイスにアクセスするユーザ アカウントを管理できます。デバイスへのログインを試みると、Cisco NX-OS によって、中央の RADIUS サーバの情報に基づいてユーザ検証が行われます。

許可は、認証されたユーザのアクセス許可範囲を意味しています。ユーザに割り当てたロールは、ユーザにアクセスを許可する実デバイスのリストとともに、RADIUS サーバに保管できます。ユーザが認証されると、デバイスはRADIUS サーバを参照して、ユーザのアクセス範囲を決定します。

アカウンティングは、デバイスの管理セッションごとに保管されるログ情報を意味しています。この情報を使用して、トラブルシューティングおよびユーザアカウンタビリティのレポートを生成できます。アカウンティングは、ローカルまたはリモートで実装できます（RADIUS を使用して）。

次に、アカウンティング ログ エントリを表示する例を示します。

```
switch# show accounting log
Sun May 12 04:02:27 2007:start:/dev/pts/0_1039924947:admin
Sun May 12 04:02:28 2007:stop:/dev/pts/0_1039924947:admin:vsh exited normally
Sun May 12 04:02:33 2007:start:/dev/pts/0_1039924953:admin
Sun May 12 04:02:34 2007:stop:/dev/pts/0_1039924953:admin:vsh exited normally
Sun May 12 05:02:08 2007:start:snmp_1039928528_172.22.95.167:public
Sun May 12 05:02:08 2007:update:snmp_1039928528_172.22.95.167:public:Switchname
```

（注） アカウンティング ログは、各セッションの最初と最後（開始と終了）だけを表示します。

syslog の使用

システム メッセージ ロギング ソフトウェアを使用して、メッセージをログ ファイルに保存するか、または他のデバイスに転送します。この機能では、次のことができます。

- モニタリングおよびトラブルシューティングのためのログ情報の記録
- キャプチャするログ情報のタイプの選択
- キャプチャするログ情報の宛先の選択

syslogを使用してシステム メッセージを時間順にローカルに保存したり、中央のsyslogサーバにこの情報を送信したりできます。syslogメッセージをコンソールに送信してすぐに使用することもできます。これらのメッセージの詳細は、選択した設定によって異なります。

syslogメッセージは、重大度に応じて、debugからcriticalまでの7つのカテゴリに分類されます。デバイス内の特定のサービスについて、レポートされる重大度を制限できます。たとえば、OSPFサービスのデバッグイベントのみを報告し、BGPサービスのすべての重大度レベルのイベントを記録することができます。

ログ メッセージは、システム再起動後には消去されています。ただし、重大度がCritical以下（レベル0、1、2）の最大100個のログ メッセージはNVRAMに保存されます。このログは、**show logging nvram** でいつでも表示できます。 コマンドを使用します。

ログ レベル

Cisco NX-OS では、次のロギング レベルがサポートされています。

- 0-emergency（緊急）
- 1-alert（警報）
- 2-critical（重大）
- 3-error（エラー）
- 4-warning（警告）
- 5-notification（通知）
- 6-informational（情報）
- 7-debugging（デバッグ）

デフォルトでは、デバイスにより、正常だが重要なシステム メッセージがログ ファイルに記録され、それらのメッセージがシステム コンソールに送信されます。ユーザは、ファシリティ タイプおよび重大度に基づいて、保存するシステム メッセージを指定できます。リアルタイムのデバッグおよび管理を強化するために、メッセージにはタイム スタンプが付加されます。

Telnet または SSH へのロギングのイネーブル化

システム ロギング メッセージは、デフォルトまたは設定済みのロギング ファシリティおよび重大度の値に基づいてコンソールに送信されます。

- コンソールのロギングをディセーブルにするには、**no logging console** コマンドをコンフィギュレーション モードで使用します。
- Telnet または SSH のロギングを有効にするには、 **terminal monitor** コマンドを実行します。
- コンソールセッションへのロギングをディセーブルまたはイネーブルにすると、その状態は、それ以後のすべてのコンソールセッションに適用されます。ユーザがセッションを終了して新規のセッションに再びログインした場合、状態は維持されています。ただし、TelnetセッションまたはSSHセッションへのロギングをイネーブルまたはディセーブルにすると、その状態はそのセッションだけに適用されます。ユーザがセッションを終了したあとは、その状態は維持されません。

この項で説明している **no logging console** コマンドは、コンソールロギングをディセーブルにし、デフォルトでイネーブルになっています。

```
switch(config)# no logging console
```

この項で説明している **terminal monitor** コマンドは、Telnet または SSH のロギングを有効にし、デフォルトではディセーブルになっています。

```
switch# terminal monitor
```

syslog の設定の詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。

SPAN の使用

スイッチドポートアナライザ（SPAN）ユーティリティを使って、詳細なトラブルシューティングの実行または特定のアプリケーションホストからトラフィックのサンプルを取得し、プロアクティブなモニタリングと分析を行うことができます。

デバイス設定を修正しても解決できない問題がネットワークにある場合は、通常、プロトコルレベルを調べる必要があります。**debug** コマンドを使用すれば、 エンド ノードとデバイス間の制御トラフィックを調べることができます。ただし、特定のエンドノードを発信元または宛先とするすべてのトラフィックに焦点を当てる必要がある場合は、プロトコルアナライザを使用してプロトコル トレースをキャプチャします。

プロトコルアナライザを使用するには、分析対象のデバイスへのラインにアナライザを挿入する必要があります。このとき、デバイスとの入出力（I / O）は中断されます。

イーサネット ネットワークでは、SPANユーティリティを使用してこの問題を解決できます。SPAN を使用すると、すべてのトラフィックのコピーを取得して、デバイス内の別のポートに

転送できます。このプロセスはどの接続デバイスも中断せず、ハードウェア内で実施されるので不要な CPU 負荷を防ぎます。

SPAN を使用すると、デバイス内で独立した SPAN セッションが作成されます。フィルタを適用して、受信したトラフィックまたは送信したトラフィックのみをキャプチャできます。

SPAN ユーティリティを開始するには、**span session** *span-num* コマンドを使用します。ここで *span-num* は特定の SPAN セッションを示します。このコマンドを入力すると、サブメニューが表示され、宛先インターフェイスと送信元 VLAN を設定できます。

```
switch2# config terminal
switch2(config)# span session 1 <<=== Create a span session
switch2(config-span)# source interface e1/8 <<=== Specify the port to be spanned
switch2(config-span)# destination interface e1/3 <<==== Specify the span destination
port
switch2(config-span)# end
switch2# show span session 1
Session 1 (active)
Destination is e1/3
No session filters configured
Ingress (rx) sources are
e1/8,
Egress (tx) sources are
fe1/8,
```

SPAN の設定の詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。

SPAN 整合性チェッカー

SPAN 整合性チェッカーは、スーパーバイザ、ラインカード、およびハードウェア テーブルのプログラムと整合性設定のチェックを実行します。スイッチで SPAN を設定すると、その状態がソフトウェア、ストレージ、ライン カード、およびハードウェア テーブルにプログラムされます。これらの状態が互いに同期していない場合、SPAN セッションは失敗します。SPAN 整合性チェッカーは、即座に修正できる SPAN セッションの不整合を識別するのに役立ちます。

cc_monitor_session.py は、SPAN 整合性チェッカーの Python スクリプトです。この Python スクリプトは、スーパーバイザ、ライン カード、およびハードウェア テーブルの状態を取得し、すべての状態が互いに同期しているかどうかを確認します。

次に、SPAN 整合性チェッカーの CLI を示します。

```
show consistency-checker monitor session {<session-id> | all}
```

この CLI は、バックエンドで Python スクリプトを実行し、SPAN 整合性チェッカーの出力を表示します。出力は次のとおりです。

```
switch# show consistency-checker monitor session 1
Monitor Consistency Check : PASSED
```

sFlow を使用

サンプリングされた Flow（sFlow）を使用すると、スイッチやルータを含むデータネットワーク内のリアルタイムトラフィックをモニターできます。sFlow では、トラフィックをモニタするためにスイッチとルータ上の sFlow エージェント ソフトウェアでサンプリング メカニズムを使用して、サンプル データを中央のデータ コレクタに転送します。sFlow の詳細については、RFC 3176 を参照してください。

Cisco NX-OS ソフトウェアに組み込まれている sFlow エージェントは、サンプリングされるパケットのデータ ソースに関連付けられたインターフェイス カウンタを定期的にサンプリングまたはポーリングします。

sFlow 構成の詳細については、『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』を参照してください。

sFlow 整合性チェッカー

sFlow 整合性チェッカーは、スーパーバイザーとライン カード ハードウェア テーブルのプログラムと整合性構成のチェックを実行します。スイッチで sFlow を構成すると、その状態がソフトウェア、ストレージ、ライン カード、およびハードウェア テーブルにプログラムされます。しかし、Cisco Nexus 9808 スイッチでは、整合性チェッカーは、スーパーバイザーとライン カード ハードウェア 抽象化レイヤーのプログラムと整合性構成のチェックを実行します。スイッチ上で sFlow を構成中、状態が互いに同期していない場合、SPAN セッションは失敗します。sFlow 整合性チェッカーは、即座に修正できる sFlow セッションの不整合を識別するのに役立ちます。

sFlow 整合性チェッカーを使用して、sFlow スーパーバイザプロセスの構成の整合性を検証できます。

（注） sFlow 整合性チェッカーは、sFlow プロセスのデータ送信元に関連する sFlow 構成情報のみを検証します。

次に、sFlow 整合性チェッカーのコマンドを示します。

```
switch(config)# show consistency-checker sflow
```

次に、出力例を示します。

```
switch(config)# show consistency-checker sflow
SFLOW CC validation start:
passed for interface ethernet 1/15
Consistency checker passed for SFLOW
```

ブルー ビーコン機能の使用

一部のプラットフォームでは、プラットフォームの LED を点滅させることができます。この機能は、ローカル管理者がトラブルシューティングや交換のためにハードウェアを迅速に識別できるように、ハードウェアをマークするのに便利な方法です。

ハードウェア エンティティの LED を点滅させるには、次のコマンドを使用します。

コマンド	目的
blink chassis	シャーシLEDを点滅させます。
blink fan *number*	ファン LED の 1 つを点滅させます。
blink module *slot*	選択したモジュールの LED を点滅させます。
blink powersupply *number*	電源 LED の 1 つを点滅させます。

watch コマンドの使用

watch コマンドを使用すると、Cisco NX-OS CLI コマンド出力または UNIX コマンド出力を更新し、監視することを許可します（**run bash** コマンド コマンドを通して）。

次のコマンドを使用します。

watch [**differences**] [**interval** *seconds*] *command*watch

- **differences**：コマンド出力の違いを強調表示します。
- **interval** *seconds*：コマンド出力を更新する頻度を指定します。範囲は 0 ～ 2147483647 秒です。
- *command*：監視するコマンドを指定します。

次に、**watch** コマンドを使用して **show interface eth1/15 counters** コマンドの出力を毎秒更新し、相違点を強調表示する例を示します。

```
switch# watch differences interval 1 show interface eth1/15 counters

Every 1.0s:  vsh -c "show interface eth1/15 counters"      Mon Aug 31 15:52:53 2015


--------------------------------------------------------------------------------
Port                                   InOctets                      InUcastPkts
--------------------------------------------------------------------------------
Eth1/15                                  583736                                0

--------------------------------------------------------------------------------
Port                                 InMcastPkts                     InBcastPkts
--------------------------------------------------------------------------------
Eth1/15                                    2433                                0

--------------------------------------------------------------------------------
Port                                  OutOctets                    OutUcastPkts
```

```
--------------------------------------------------------------------------------
Eth1/15                                 5247672                               0

--------------------------------------------------------------------------------
Port                               OutMcastPkts                     OutBcastPkts
--------------------------------------------------------------------------------
Eth1/15                                   75307                               0
```

トラブルシューティングのツールと方法論の追加参照

関連資料

関連項目	マニュアル タイトル
システム管理ツール	『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』
MIB	『*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*』

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。