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。

```
2013 Nov 1 16:31:39 switch vshd: CC_LINK_STATE:
Consistency Check: PASSED
```

Cisco NX-OS は、次の整合性チェッカーをサポートします。

- **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。
- **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。
- **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。
- **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。
- **show consistency-checker membership vlan** *vlan-id* {**native-vlan** | **private-vlan interface** {**ethernet** *slot/port* | **port-channel** *number* | **native-vlan**}} [**brief** | **detail**]：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。

- **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 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の IPv4 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） 現在、このコマンドは、IPv4 および IPv6 ACL を検証せず、サブインターフェイスで検証せずに、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。

- **show consistency-checker stp-state vlan** *vlan-id*：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。

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。**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。

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。

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、 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。

コマンド	目的
[**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。

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。

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。
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。

コマンド	目的
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 interfacefront-panelinbandinband-ininband-outmgmtport-channelvlanlimit-captured-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、0〜500,000 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。
{|||||| } **ethanalyzer local interfacefront-panelinbandinband-ininband-outmgmtport-channelvlanlimit-frame-size**	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 interfacefront-panelinbandinband-ininband-outmgmtport-channelvlancapture-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 interfacefront-panelinbandinband-ininband-outmgmtport-channelvlandisplay-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 interfacefront-panelinbandinband-ininband-outmgmtport-channelvlanwrite**	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。

コマンド	目的
{| | | | | | } **ethanalyzer local interface** **front-panel** **inband** **inband-in** **inband-out** **mgmt** **port-channel** **vlan** **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** **front-panel** **inband** **inband-in** **inband-out** **mgmt** **port-channel** **vlan** **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** **front-panel** **inband** **inband-in** **inband-out** **mgmt** **port-channel** **vlan** **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。
{| | | | | | } **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 write bootflash:***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
```

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。

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 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。

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。