APPENDIX B

トラブルシューティングのツールおよび方法

この付録では、Cisco NX-OS で使用可能なトラブルシューティングのツールおよび方法について説明します。この章で説明する内容は、次のとおりです。

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

Command-Line Interface（CLI; コマンドライン インターフェイス）では、ローカル コンソールを使用するか、またはリモートから Telnet や Secure Shell（SSH; セキュア シェル）セッションを使用して、Cisco NX-OS の設定および監視を実行できます。CLI のコマンド構造は Cisco IOS ソフトウェアと似ており、コンテキスト ヘルプ、**show** コマンド、マルチユーザ サポート、ロールベースのアクセス制御を使用できます。

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、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。エラー コードに関する詳細を確認するには、**show system error-id** コマンドを使用します。

```
switch# copy running-config startup-config
[########################################] 100%
2008 Jan 16 09:59:29 zoom %$ VDC-1 %$ %BOOTVAR-2-AUTOCOPY_FAILED: Autocopy of file
/bootflash/n7000-s1-dk9.4.0.0.837.bin.S8 to standby failed, error=0x401e0008

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

- **show platform** — ルート転送、Quality of Service（QoS）、Access Control List（ACL; アクセス制御リスト）情報を含むプラットフォーム別の情報を提供します。

コンフィギュレーション ファイル

コンフィギュレーション ファイルには、Cisco NX-OS デバイスで機能を設定するための Cisco NX-OS コマンドが含まれています。Cisco NX-OS には、実行コンフィギュレーションとスタートアップ コンフィギュレーションという 2 つのタイプのコンフィギュレーション ファイルがあります。デバイスは、デバイスの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われた現在の変更が保存されます。コンフィギュレーションを変更する前に、コンフィギュレーション ファイルのバックアップ バージョンを作成する必要があります。コンフィギュレーション ファイルをリモート サーバにバックアップすることも（『*Cisco NX-OS Fundamentals Configuration Guide, Release 4.0*』のコンフィギュレーション ファイルの情報を参照）、問題発生時のロールバック用のコンフィギュレーション ファイルのチェックポイント コピーを作成することも（『*Cisco NX-OS System Management Configuration Guide, Release 4.0*』のロールバック機能を参照）できます。

Cisco NX-OS の機能は、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できます。まれにですが、これらのロックを解除できないこともあります。スタートアップ コンフィギュレーション ファイルにロックが残っているかどうかを判別するには、**show system internal sysmgr startup-config locks** コマンドを使用します。これらのロックを解除するには、**system startup-config unlock** コマンドを使用します。

CLI によるデバッグ

Cisco NX-OS では、ネットワークのトラブルシューティングを行うための多数のデバッグ機能セットがサポートされています。CLI を使用して、各機能のデバッグ モードをイネーブルにすると、制御プロトコル交換のリアルタイム更新動作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付加され、発生時刻順に表示されます。デバッグ機能へのアクセスは、CLI のロール機構を使用して制限し、ロール単位でアクセスを分割できます。**debug** コマンドでは、リアルタイムの情報が表示されますが、**show** コマンドを使用すると、リアルタイム情報とともに履歴情報も表示できます。

注意　一部の **debug** コマンドはネットワークのパフォーマンスに影響を与える可能性があるため、**debug** コマンドはシスコ テクニカル サポート担当者の指示に従って使用してください。

（注）　デバッグ メッセージは、特殊なログ ファイルに記録できます。ログ ファイルに記録した方が、デバッグ出力をコンソールに送信するよりも安全で、処理も簡単です。

? オプションを使用すると、各機能で利用できるオプションを表示できます。各コマンド入力には、実デバッグ出力のほかにログ 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動作のタイムスタンプ付きアカウントが表示されます。

デバッグ機能を使用すると、イベント、内部メッセージ、およびプロトコル エラーをトラッキングできます。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には注意が必要です。オプションによっては、コンソールへの出力メッセージが大量に生成されるためにデバイスにアクセスできなくなったり、また CPU に大きな負荷がかかるイベントが生成されてパフォーマンスが著しく低下したりすることがあります。

（注）　**debug-filter** CLI コマンドを使用すると、不要なデバッグ情報をフィルタリングして取り除くことができます。

（注）　**debug** コマンドを入力する前に 2 番めの Telnat または SSH セッションを開くことを推奨します。デバッグ セッションの現在の出力ウィンドウに対する表示が速すぎて停止できない場合、2 番めのセッションを使用して **undebug all** コマンドを入力すれば、デバッグ メッセージの出力を停止できます。

ping および traceroute

（注）　ping および traceroute 機能は、接続およびパス選択に関する問題のトラブルシューティングに使用します。これらの機能を、パフォーマンスの問題の識別または解決には使用しないでください。

TCP/IP ネットワーキングに関する問題のトラブルシューティングを行う場合、最も効果的な 2 つのツールが、ping および traceroute 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 ユーティリティの動作も似ていますが、さらに、フレームが経由した宛先までの特定パスをホップ単位で判別できます。

ここで説明する内容は、次のとおりです。

ping の使用

ping コマンドを使用すると、IPv4 ルーティング型ネットワーク内の特定の宛先に対する接続および遅延を確認できます。

ping6 コマンドを使用すると、IPv6 ルーティング型ネットワーク内の特定の宛先に対する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 デバイスに短い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と、ターゲットの宛先に一連のフレームが送信されます。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。

traceroute の使用

traceroute は、次の目的で使用します。

- 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、双方向のパスがホップ単位で識別され、ホップごとにタイムスタンプが付加されます。traceroute を使用すると、発信スイッチと宛先に最も近いスイッチ間のパスに沿って、ポートの接続をテストできます。

IPv4 ネットワークでは **traceroute** コマンドを使用し、IPv6 ネットワークでは **traceroute6** コマンドを使用します。

宛先に到達できない場合には、パス検出が開始され、障害ポイントまでのパスがトラッキングされます。

プロセスおよび CPU のモニタリング

ここで説明する内容は、次のとおりです。

show processes CLI コマンドの使用

show processes コマンドを使用すると、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（例 B-1 を参照）。このコマンドの出力には、次の情報が表示されます。

- 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された回数（または再開）
- 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、次のように示されます。

- 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 = 存在しない（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 ステートは、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。

例 B-1　show processes コマンド

```
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   Process
-----  -----  --------  -----------  ----  -------------
    1      S  b7f9e468            1     -  init
    2      S         0            1     -  migration/0
    3      S         0            1     -  ksoftirqd/0
    4      S         0            1     -  desched/0
    5      S         0            1     -  migration/1
    6      S         0            1     -  ksoftirqd/1
    7      S         0            1     -  desched/1
    8      S         0            1     -  events/0
    9      S         0            1     -  events/1
   10      S         0            1     -  khelper
   15      S         0            1     -  kthread
   24      S         0            1     -  kacpid
  101      S         0            1     -  kblockd/0
  102      S         0            1     -  kblockd/1
  115      S         0            1     -  khubd
  191      S         0            1     -  pdflush
  192      S         0            1     -  pdflushn
...
```

show processes cpu CLI コマンドの使用

show processes cpu コマンドを使用すると、CPU 使用率を表示できます（例 B-2 を参照）。このコマンドの出力には、次の情報が表示されます。

- 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 = 開始された各プロセスの CPU 時間の平均（ミリ秒単位）
- 1Sec = 最近の 1 秒間における CPU 使用率（パーセント表示）

例 B-2　show processes cpu コマンド

```
switch# show processes cpu

PID    Runtime(ms)  Invoked    uSecs  1Sec   Process
-----  -----------  --------  -----  -----  -----------
    1          922  4294967295      0      0  init
    2          580    377810      1      0  migration/0
    3          889   3156260      0      0  ksoftirqd/0
    4         1648    532020      3      0  desched/0
    5          400    150060      2      0  migration/1
    6         1929   2882820      0      0  ksoftirqd/1
    7         1269    183010      6      0  desched/1
    8         2520  47589180      0      0  events/0
    9         1730   2874470      0      0  events/1
   10           64    158960      0      0  khelper
   15            0    106970      0      0  kthread
   24            0     12870      0      0  kacpid
  101           62   3737520      0      0  kblockd/0
  102           82   3806840      0      0  kblockd/1
  115            0     67290      0      0  khubd
  191            0      5810      0      0  pdflush
  192          983   4141020      0      0  pdflush
  194            0      5700      0      0  aio/0
  193            0      8890      0      0  kswapd0
  195            0      5750      0      0  aio/1
...
```

show system resource CLI コマンドの使用

show system resources コマンドを使用すると、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（例 B-3 を参照）。このコマンドの出力には、次の情報が表示されます。

- Load は、実行中プロセスの数として定義されます。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の数、およびコマンドの実行時に実際に稼働していたプロセスの数が表示されます。
- 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 KB 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。

例 B-3　show system resources コマンド

```
switch# show system resources
Load average:   1 minute: 0.30   5 minutes: 0.34   15 minutes: 0.28
Processes   :   606 total, 2 running
CPU states  :   0.0% user,   0.0% kernel,   100.0% idle
Memory usage:   2063268K total,   1725944K used,    337324K free
                   2420K buffers,  857644K cache
```

オンボード障害ログ機能の使用

Cisco NX-OS には、障害データを永続的ストレージ上のログに記録する機能があります。このデータは、分析の目的で取得して表示できます。この Onboard Failure Logging（OBFL; オンボード障害ログ）機能では、障害情報および環境情報をモジュール上の不揮発性メモリに保管します。この情報は、故障したモジュールの分析に役立ちます。

OBFL 機能によって記録されるデータは、次のとおりです。

- 初期電源投入の時間
- シャーシ内にあるカードのスロット番号
- カードの初期温度
- ファームウェア、BIOS、FPGA 、および ASIC のバージョン
- カードのシリアル番号
- クラッシュに対するスタック 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 限定の履歴情報
- ASIC 割り込みおよびエラー統計情報の履歴
- ASIC レジスタのダンプ

OBFL の設定の詳細については、『*Cisco NX-OS System Management Configuration Guide, Release 4.0*』を参照してください。

GOLD 診断の使用

Generic Online Diagnostics（GOLD; 汎用オンライン診断）は、シスコの全プラットフォームの診断処理に共通のフレームワークを定義します。GOLD により、ハードウェア コンポーネントがチェックされ、システムのデータ プレーンとコントロール プレーンが正常に動作しているかどうかが確認されます。テストによっては、システムの起動時に実施されたり、システム稼働中に実施されたりします。

ブート モジュールは一連のチェックを受けてからオンラインになります。これにより、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でき、障害のあるモジュールが稼働中のネットワークに入り込むことを防止できます。

障害の診断は、システムの稼働中（ランタイム）にも実施されます。一連の診断チェックを設定すると、オンライン システムの状態を判別できます。ただし、ディスラプティブ（中断を伴う）テストと、ノンディスラプティブ（中断を伴わない）テストを区別して実行する必要があります。ノンディスラプティブ テストはバックグラウンドで実行されるため、システム データやコントロール プレーンに影響を与えませんが、ディスラプティブ テストは稼働中のパケット フローに影響を及ぼします。ディスラプティブ テストは、特別にメンテナンス時間枠を設けて計画的に実施する必要があります。**show diagnostic content module** CLI コマンドの出力には、ディスラプティブ テスト、ノンディスラプティブ テストなど、テストの属性が表示されます。

ランタイム診断チェックは、特定の時間に実行、または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、ノンディスラプティブ テストで、システムの稼働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、稼働中のネットワーク環境でハードウェア障害をプロアクティブに検出し、管理者に障害を通知することです。

GOLD では、すべてのテストの診断結果と詳細な統計情報（最後の実行時刻、最初と最後のテストパス時刻、最初と最後のテスト失敗時刻、総実行回数、総失敗回数、失敗連続回数、およびエラーコード）が収集されます。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特定するのに役立ちます。**show diagnostic result** コマンドを使用すると、診断結果を表示できます。

GOLD の設定の詳細については、『*Cisco NX-OS System Management Configuration Guide, Release 4.0*』を参照してください。

EEM の使用

Embedded Event Manager（EEM; 組み込みイベント マネージャ）は、重要なシステム イベントを監視し、ポリシー セットを通じてこれらのイベントに対処できるようにするポリシーベースのフレームワークです。ポリシーは事前にプログラミングされたスクリプトです。発生したイベントに応じて呼び出す処理をこのスクリプトに定義し、ロードすることができます。スクリプトでは、カスタム Syslog または SNMP トラップの生成、CLI コマンド呼び出し、フェールオーバーの強制をはじめ、さまざまな処理を生成できます。

EEM の設定の詳細については、『*Cisco NX-OS System Management Configuration Guide, Release 4.0*』を参照してください。

Ethanalyzer の使用

Ethanalyzer は、Wireshark（旧称 Ethereal）オープン ソース コードに基づく Cisco NX-OS プロトコル アナライザ ツールです。Ethanalyzer は、パケットのキャプチャとデコード用の Wireshark のコマンド ライン バージョンです。Ethanalyzer は、ネットワークのトラブルシューティングおよびコントロール プレーン トラフィックを分析する場合に使用します。

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	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、詳細なプロトコル情報を提供します。
ethanalyzer local interface brief	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、プロトコル情報の概要を提供します。
ethanalyzer local interface limit-captured-frames	キャプチャするフレーム数を制限します。
ethanalyzer local interface limit-frame-size	キャプチャするフレームの長さを制限します。
ethanalyzer local interface capture-filter	キャプチャするパケットのタイプをフィルタします。
ethanalyzer local interface display-filter	表示するキャプチャ済みパケットのタイプをフィルタします。
ethanalyzer local interface decode-internal	Cisco NX-OS の内部フレーム ヘッダをデコードします。 **（注）** NX-OS Ethanalyzer の代わりに Wireshark を使用してデータを分析するときは、このオプションを使用しないでください。
ethanalyzer local interface write	キャプチャしたデータをファイルに保存します。
ethanalyzer local interface read	キャプチャしたデータのファイルを開き、分析します。

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。詳細については、次の URL を参照してください。

http://www.tcpdump.org/tcpdump_man.html

表示フィルタの構文の詳細については、次の URL を参照してください。

http://wiki.wireshark.org/DisplayFilters

管理インターフェイス上に、キャプチャしたデータ（4 パケットに制限）を表示する例を示します。

```
switch(config)# ethanalyzer local interface mgmt brief limit-captured-frames 4
Capturing on eth1
2008-02-18 13:21:21.841182 172.28.230.2 -> 224.0.0.2    HSRP Hello (state Stand
y)
2008-02-18 13:21:21.842190 10.86.249.17 -> 172.28.231.193 TCP 4261 > telnet [AC
] Seq=0 Ack=0 Win=64475 Len=0
2008-02-18 13:21:21.843039 172.28.231.193 -> 10.86.249.17 TELNET Telnet Data ..
2008-02-18 13:21:21.850463 00:13:5f:1c:ee:80 -> ab:00:00:02:00:00 0x6002 DEC DN
 Remote Console
4 packets captured
```

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する例を示します。

```
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: Feb 18, 2008 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/broadca
st)
        .... ..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: 0
    Authentication Data: Default (cisco)
    Virtual IP Address: 172.28.230.1 (172.28.230.1)

1 packets captured
```

表示フィルタを利用して、アクティブな HSRP ステートを持つ HSRP パケットのみを表示する例を示します。

```
switch(config)# ethanalyzer local interface mgmt brief display-filter "hsrp.stat
e==Active" limit-captured-frames 2
Capturing on eth1
2008-02-18 14:35:41.443118 172.28.230.3 -> 224.0.0.2    HSRP Hello (state Active
)
2008-02-18 14:35:44.326892 172.28.230.3 -> 224.0.0.2    HSRP Hello (state Active
)
2 packets captured
```

Wireshark の詳細については、次の URL を参照してください。

http://www.wireshark.org/docs/

DCNM ツール

Cisco DCNM は、サポートされている各機能に関するイベントと統計情報を収集します。

DCNM の詳細については、『*Cisco DCNM Fundamentals Configuration Guide, Release 4.0*』を参照してください。

SNMP および RMON のサポート

Cisco NX-OS は、Management Information Base（MIB）および通知（トラップおよびインフォーム）を含む、SNMP v1、v2、および v3 を包括的にサポートしています。

SNMP 標準により、異なる MIB をサポートしているサードパーティ製のアプリケーションを使用して、Cisco NX-OS の管理および監視を行うことができます。

SNMP v3 は、拡張セキュリティを提供します。各デバイスでは、SNMP サービスを選択的にイネーブルまたはディセーブルに設定できます。また、各デバイスを SNMP v1 および v2 要求の処理方式で設定できます。

Cisco NX-OS では、Remote Monitoring（RMON; リモート モニタリング）アラームおよびイベントもサポートしています。RMON アラームおよびイベントは、しきい値の設定や、ネットワーク動作の変更に基づく通知の送信などのメカニズムを提供します。

AlarmGroup では、アラームを設定することができます。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 使用率のレベルを指定して、RMON アラームを設定できます。*EventGroup* では、アラーム条件に基づいて実行される動作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あります。

SNMP および RMON の設定の詳細については、『*Cisco NX-OS System Management Configuration Guide, Release 4.0*』を参照してください。

RADIUS の使用

RADIUS 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、アトリビュートまたは証明書を交換するためのプロトコルです。これらのアトリビュートは、次の 3 つの Class of Service（CoS; サービス クラス）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共に、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識別します。

アカウンティングは、スイッチ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、（RADIUS を使用して）ローカルまたはリモートで実行できます。

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 Dec 15 04:02:27 2007:start:/dev/pts/0_1039924947:admin
Sun Dec 15 04:02:28 2007:stop:/dev/pts/0_1039924947:admin:vsh exited normally
Sun Dec 15 04:02:33 2007:start:/dev/pts/0_1039924953:admin
Sun Dec 15 04:02:34 2007:stop:/dev/pts/0_1039924953:admin:vsh exited normally
Sun Dec 15 05:02:08 2007:start:snmp_1039928528_172.22.95.167:public
Sun Dec 15 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（開始時と終了時）のみを表示します。

Syslog の使用

システム メッセージ ロギング ソフトウェアでは、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。

- モニタおよびトラブルシューティングのためのログ情報を記録
- 取り込まれるログ情報のタイプを選択
- 取り込まれるログ情報の宛先を選択

Syslog を使用すると、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することができます。すぐに使用する場合には、Syslog メッセージをコンソールに出力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、デバッグからクリティカル イベントまで、7 つのカテゴリに分類されます。デバイス内の特定サービスについて、レポートする重大度レベルを制限できます。たとえば、OSPF サービスについてはデバッグ イベントだけをレポートし、BGP サービスについてはすべての重大度レベルのイベントを記録するといった設定ができます。

ログ メッセージは、システム再起動の全体にわたって保存されるものではありません。ただし、重大度が critical 以上（レベル 0、1、および 2）のログ メッセージは、最大 100 個まで NVRAM に保存されます。このログは、**show logging nvram** コマンドを使用していつでも表示できます。

ここで説明する内容は、次のとおりです。

- ログ レベル（p.B-14）
- Telnet または SSH へのロギングのイネーブル化（p.B-15）

ログ レベル

Cisco NX-OS では、次のログ 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、標準的かつ重要なシステム メッセージがログ ファイルに記録され、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機能を強化するために、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。

ユーザは、コンソールへのロギングをディセーブルにすること、または特定の Telnet や SSH セッションへのロギングをイネーブルにすることができます。

- コンソールへのロギングをディセーブルにするには、設定モードで **no logging console** コマンドを使用します。
- Telnet または SSH へのロギングをイネーブルにするには、EXEC モードで **terminal monitor** コマンドを使用します。

（注）　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、そのステートが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、ステートは維持されます。ただし、Telnet または SSH 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、そのステート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、そのステートは維持されません。

no logging console コマンド（例 B-4 を参照）は、コンソールへのロギングをディセーブルにします。デフォルト設定はイネーブルです。

例 B-4　no logging console コマンド

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

terminal monitor コマンド（例 B-5 を参照）は、Telnet または SSH へのロギングをイネーブルにします。デフォルト設定はディセーブルです。

例 B-5　terminal monitor コマンド

```
switch# terminal monitor
```

Syslog の設定の詳細については、『*Cisco NX-OS System Management Configuration Guide, Release 4.0*』を参照してください。

SPAN の使用

Switched Port Analyzer（SPAN; スイッチド ポート アナライザ）ユーティリティを使用すると、詳細なトラブルシューティングを実行すること、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をサンプリングして予防的なモニタリングおよび分析を実行できます。

デバイスのコンフィギュレーションを修正しても ネットワークの問題を解決できない場合には、通常、プロトコル レベルを調べる必要があります。エンド ノードとスイッチ間の制御トラフィックは、**debug** コマンドによって確認できます。ただし、ホストまたはディスクなどの特定のエンド ノードが送信または受信しているすべてのトラフィックを調べる必要がある場合には、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できます。

プロトコル アナライザを使用するには、分析対象デバイスの回線内にアナライザを挿入し、デバイスの入出力に割り込む必要があります。

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することによって、この問題を解決できます。SPAN では、すべてのトラフィックをコピーして、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、どの接続デバイスにも割り込まず、CPU に不要な負荷がかからないようにハードウェアで実行されます。

SPAN では、デバイス内で最大 16 の個別の SPAN セッションを作成できます。各セッションに、最大 4 つの個別の送信元と、1 つの宛先ポートを設定します。また、フィルタを適用することにより、受信トラフィックまたは送信トラフィックだけをキャプチャできます。特定の VLAN からのトラフィックをキャプチャすることもできます。

SPAN ユーティリティを起動するには、**span session** *session_num* コマンドを使用し、*session_num* に各 SPAN セッションの識別番号を指定します。このコマンドを入力すると、宛先インターフェイスおよび送信元の VLAN またはインターフェイスを設定できるサブメニューが表示されます。

```
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 NX-OS System Management Configuration Guide, Release 4.0*』を参照してください。

Blue Beacon 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。ローカルの管理者が、トラブルシューティングや交換を行うハードウェアをすぐに識別できるように、ハードウェア コンポーネントの LED を点滅させておくと便利です。

ハードウェアの 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	シャーシの LED を点滅させます。
blink fan *number*	ファンの LED 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*	電源の LED を点滅させます。
blink xbar *number*	クロスバー モジュールの LED の 1 つを点滅させます。

モジュールのシングル ポート LED を点滅させるには、インターフェイス設定モードで次のコマンドを使用します。

コマンド	目的
beacon	インターフェイスの LED を点滅させます。