

SNMP を使用してハングしている TCP 接続を検出しクリアする方法

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[MIB オブジェクトの詳細：オブジェクト識別子 \(OID\) を含む](#)

[TCP 接続のハングを検出するために SNMP を使用する](#)

[要約](#)

[手順ごとの説明](#)

[ハングしている TCP 接続をクリアするために SNMP を使用する](#)

[手順ごとの説明](#)

[MIB オブジェクトの詳細情報](#)

[ハングした TCP 接続を検出してクリアする PERL スクリプト](#)

[関連情報](#)

概要

このドキュメントでは、Cisco IOS デバイスでハングした TCP 接続を検出してクリアするために Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) を使用する方法について説明します。また、この目的のために使用する SNMP オブジェクトについても説明します。

「[ハングした TCP 接続を検出してクリアする PERL スクリプト](#)」というタイトルのセクションでは、これらの手順を実装している PERL スクリプトへのリンクを提供しています。

前提条件

要件

このドキュメントの読者は次のトピックについての専門知識を有している必要があります。

- Cisco デバイス上で TCP 接続情報を表示する方法の理解
- SNMP の `walk`、`get`、`get-next`、`set` コマンドの一般的な使用方法
- Cisco デバイス上での SNMP の設定方法の理解

使用するコンポーネント

このドキュメントは、TCP-MIB モジュールと [CISCO-TCP-MIB](#) モジュールをサポートしている IOS ソフトウェアを稼働している Cisco ルータとスイッチに適用されます。


注：CISCO-TCP-MIBモジュールは、デフォルトではNET-SNMPにはロードされません。MIB モジュールがシステムにロードされない場合、オブジェクトの名前ではなく OID を使用してオブジェクトを参照する必要があります。

このドキュメントの情報は、すべての IOS ソフトウェアとハードウェアのバージョンに基づいています。

情報は、NET-SNMP の次のバージョンに基づいています。

• NET-SNMP バージョン 5.1.2 は、<http://www.net-snmp.org/> で入手できます。 

PERL スクリプトは次の PERL バージョンでテストされました。

- FreeBSD での 5.005_03
- Solaris 5.8 での 5.8.0
- 5.005_02 : Microsoft Windows 2000 上で CiscoWorks SNMS の一部として出荷
- Microsoft Windows 2000上のActivePerl 5.8.4。
<http://www.activestate.com/Products/ActivePerl/>から入手 **できます** 

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、「[シスコテクニカルティップスの表記法](#)」を参照してください。

背景説明

MIB オブジェクトの詳細：オブジェクト識別子 (OID) を含む

使用するオブジェクトは次のとおりです。

[CISCO-TCP-MIB](#) モジュールからの場合：

- [ciscoTcpConnInBytes](#)、OID .1.3.6.1.4.1.9.9.6.1.1.1.1この接続への入力バイト数。
- [ciscoTcpConnInPkts](#)、OID 1.3.6.1.4.1.9.9.6.1.1.1.2この接続への入力パケット数。
- [ciscoTcpConnOutBytes](#)、OID .1.3.6.1.4.1.9.9.6.1.1.1.3この接続からの出力バイト数。
- [ciscoTcpConnOutPkts](#)、OID .1.3.6.1.4.1.9.9.6.1.1.1.4この接続からの出力パケット数。
- [ciscoTcpConnRetransPkts](#)、OID .1.3.6.1.4.1.9.9.6.1.1.1.7この接続で再転送されるパケット数。
- [ciscoTcpConnRto](#)、OID .1.3.6.1.4.1.9.9.6.1.1.1.9この接続用の再転送タイムアウト値。

[TCP-MIB](#) モジュールからの場合：

- [tcpConnState](#)、OID .1.3.6.1.2.1.6.13.1.1この接続に対するステータス。

これらのオブジェクトの詳細については、「[MIB オブジェクトの詳細情報](#)」を参照してください。

TCP 接続のハングを検出するために SNMP を使用する

要約

次のステップを使用すると、TCP 接続がハングするかどうかを判断できます。

1. [ciscoTcpConnRetransPkts](#) オブジェクトと [ciscoTcpConnRto](#) オブジェクトがデバイスでサポートされているかどうかを確認するために、[ciscoTcpConnRto](#) に対して SNMP [get-next](#) 動作を実行して、何らかのオブジェクトが戻されることを確認します。注：両方のサポートが同時に追加されたため、1つのオブジェクトだけをチェックする必要があります。注：すべてのCiscoデバイスが最後の2つのオブジェクト([ciscoTcpConnRetransPkts](#)と[ciscoTcpConnRto](#))をサポートしているわけではありませんが、それらの使用によって検出の精度が向上する可能性があります。[ciscoTcpConnRetransPkts](#) オブジェクトと [ciscoTcpConnRto](#) オブジェクトがサポートされている場合、ステップ 2 に進みます。[ciscoTcpConnRetransPkts](#) オブジェクトと [ciscoTcpConnRto](#) オブジェクトがサポートされていない場合、ステップ 3 に進みます。
2. すべてのオブジェクトがサポートされています。それぞれの TCP 接続に対して次を確認します。[ciscoTcpConnOutBytes](#) が 0 である。[ciscoTcpConnOutPkts](#) が 0 である。[ciscoTcpConnRetransPkts](#) が 0 よりも大きい。[ciscoTcpConnRto](#) が 20,000 よりも大きい。注：検出を高速化するのであれば、20,000 を減らすことができます。接続がハングすると、Rto が 20,000 に達するまで 1 分間ほどかかります。ただし、小さな値によって、結果の正確さが低くなる可能性があります。前述のすべてが正しい場合、この TCP 接続はハングしており、クリアできます。「[ハングしている TCP 接続をクリアするために SNMP を使用する](#)」に進んでください。
3. 最初の 4 つのオブジェクトだけがサポートされています。それぞれの TCP 接続に対して次を確認します。[ciscoTcpConnInBytes](#) が 0 よりも大きい。[ciscoTcpConnInPkts](#) が 0 である。[ciscoTcpConnOutBytes](#) が 0 である。[ciscoTcpConnOutPkts](#) が 0 である。数秒待つてからもう一度オブジェクトを [get](#) して、それが確立されるプロセスでの TCP 接続ではなかったことを確認します。注：最初の2つのチェック (正の数の入力バイトですが、入力パケットはありません) は奇妙に見えるかもしれませんが、多くのデバイスとIOSバージョンに対して検証されました。注：6つのオブジェクトすべてをサポートするIOSバージョンでは、この動作が発生しない可能性があります。したがって、ステップ2のテストには、これらの最初の2つのテストは含まれません。どちらの場合もすべてのオブジェクトがテストを満たす場合、この TCP 接続はハングしており、クリアできます。「[ハングしている TCP 接続をクリアするために SNMP を使用する](#)」に進んでください。

手順ごとの説明

この例の値は次のとおりです。

- デバイス ホスト名 a = nms-7206a (すべてのプロジェクトをサポート)
- デバイス ホスト名 b = nms-1605 (最初の 4 つのオブジェクトだけをサポート)
- 読み取りコミュニティ = public
- 書き込みコミュニティ = private

コミュニティストリングとホスト名を次のコマンドで置き換えます。

1. このデバイスが [ciscoTcpConnRetransPkts](#) オブジェクトと [ciscoTcpConnRto](#) オブジェクトをサポートするかどうかを判断します。ciscoTcpConnRto に対して [SNMP get-next](#) 動作を実行します。

```
snmpgetnext -c public nms-7206a ciscoTcpConnRto
```

オブジェクトがサポートされている場合、次のような応答になります。

```
CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 =  
INTEGER: 303 milliseconds
```

注：これらのオブジェクトに使用されるインデックスは、ローカルIPアドレスの連結で、ローカルIPアドレスの14.32.100.75 (ローカルTCPポート番号2) です065、リモートIPアドレス：172.18.86.111、リモートTCPポート番号：23092。戻り値は [ciscoTcpConnRto](#) 用です。ステップ2に進みます。オブジェクトがサポートされていない場合、次のような応答になります。

```
snmpgetnext -c public nms-1605 ciscoTcpConnRto  
CISCO-FLASH-MIB::ciscoFlashDevicesSupported.0 = INTEGER: 1
```

戻り値は [ciscoTcpConnRto](#) オブジェクト用では[ありません](#)。戻される正確なオブジェクトは重要ではありません。手順3に進んでください。

2. Cisco TCP 接続テーブルで6つのオブジェクトすべてをサポートするデバイスに対する各TCP接続に関する情報をgetします。ciscoTcpConnOutBytes、[ciscoTcpConnOutPkts](#)、[ciscoTcpConnRetransPkts](#)、および [ciscoTcpConnRto](#) に対して [SNMP get-next](#) 動作を実行します。

```
snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes  
ciscoTcpConnOutPkts  
ciscoTcpConnRetransPkts  
ciscoTcpConnRto
```

次のような応答になります。

```
CISCO-TCP-MIB::ciscoTcpConnOutBytes.14.32.100.75.2065.172.18.86.111.23092 = Counter32:  
383556  
CISCO-TCP-MIB::ciscoTcpConnOutPkts.14.32.100.75.2065.172.18.86.111.23092 = Counter32: 8061  
CISCO-TCP-MIB::ciscoTcpConnRetransPkts.14.32.100.75.2065.172.18.86.111.23092 = Counter32: 2  
CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 = INTEGER: 303  
milliseconds
```

次の内容を確認します。[ciscoTcpConnOutBytes](#) が0である。[ciscoTcpConnOutPkts](#) が0である。[ciscoTcpConnRetransPkts](#) が0よりも大きい。[ciscoTcpConnRto](#) が20,000よりも大きい。**注：**検出を高速化するのであれば、20,000を減らすことができます。接続がハングすると、Rtoが20,000に達するまで1分間ほどかかります。ただし、小さな値によって、結果の正確さが低くなる可能性があります。これらのすべてが正しい場合、このTCP接続はハングしており、クリアできます。「[ハングしているTCP接続をクリアするためにSNMPを使用する](#)」に進んでください。TCP接続テーブルのwalkを続行します。これを実行するために、ハングした接続を確認するときに、次のような戻されたオブジェクトを使用して繰り返しSNMP get-next動作を実行します。

```
snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes.14.32.100.75.2065.172.18.86.111.23092  
ciscoTcpConnOutPkts.14.32.100.75.2065.172.18.86.111.23092  
ciscoTcpConnRetransPkts.14.32.100.75.2065.172.18.86.111.23092  
ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092
```

get-next 動作がこの方式でオブジェクトを返すまで、これまでのテストを使用して各エントリを確認します。

```
CISCO-TCP-MIB::ciscoTcpConnInPkts.14.32.100.75.2065.172.18.86.111.23092 = Counter32: 8097
CISCO-TCP-MIB::ciscoTcpConnElapsed.14.32.100.75.2065.172.18.86.111.23092 =
  Timeticks: (17296508) 2 days, 0:02:45.08
CISCO-TCP-MIB::ciscoTcpConnFastRetransPkts.14.32.100.75.2065.172.18.86.111.23092 =
Counter32: 0
CISCO-FLASH-MIB::ciscoFlashDevicesSupported.0 = INTEGER: 5
```

これで、このデバイスの TCP 接続をすべて確認して終了しました。

3. Cisco TCP 接続テーブルで最初の 4 つのオブジェクトだけをサポートするデバイスに対する各 TCP 接続に関する情報を **get** します。 `ciscoTcpConnInBytes`、[ciscoTcpConnInPkts](#)、[ciscoTcpConnOutBytes](#)、および [ciscoTcpConnOutPkts](#) に対して SNMP **get-next** 動作を実行します。

```
snmpgetnext -c public nms-1605 ciscoTcpConnInBytes
ciscoTcpConnInPkts
ciscoTcpConnOutBytes
ciscoTcpConnOutPkts
```

次のような応答になります。

```
CISCO-TCP-MIB::ciscoTcpConnInBytes.14.32.6.185.23.14.32.100.33.2249 = Counter32: 68
CISCO-TCP-MIB::ciscoTcpConnInPkts.14.32.6.185.23.14.32.100.33.2249 = Counter32: 12
CISCO-TCP-MIB::ciscoTcpConnOutBytes.14.32.6.185.23.14.32.100.33.2249 = Counter32: 170
CISCO-TCP-MIB::ciscoTcpConnOutPkts.14.32.6.185.23.14.32.100.33.2249 = Counter32: 17
```

次の内容が正しいことを確認します。 [ciscoTcpConnInBytes](#) が 0 よりも大きい。

[ciscoTcpConnInPkts](#) が 0 である。 [ciscoTcpConnOutBytes](#) が 0 である。

[ciscoTcpConnOutPkts](#) が 0 である。数秒待って、もう一度オブジェクトを **get** します。確立される処理でそれが TCP 接続ではなかったことを確認します。前述のすべてが正しい場合、この TCP 接続はハングしており、クリアできます。「[ハングしている TCP 接続をクリアするために SNMP を使用する](#)」に進んでください。TCP 接続テーブルの **walk** を続行します。これを実行するために、ハングした接続を確認するときに、次のような戻されたオブジェクトを使用して繰り返し SNMP **get-next** 動作を実行します。

```
snmpgetnext -c public nms-1605 ciscoTcpConnInBytes.14.32.6.185.23.14.32.100.33.2249
ciscoTcpConnInPkts.14.32.6.185.23.14.32.100.33.2249
ciscoTcpConnOutBytes.14.32.6.185.23.14.32.100.33.2249
ciscoTcpConnOutPkts.14.32.6.185.23.14.32.100.33.2249
```

get-next 動作がこの方式でオブジェクトを返すまで、これまでのテストを使用して各エントリを確認します。

```
CISCO-TCP-MIB::ciscoTcpConnOutBytes.14.32.6.185.23.14.32.100.33.4184 = Counter32: 170
CISCO-TCP-MIB::ciscoTcpConnOutPkts.14.32.6.185.23.14.32.100.33.4184 = Counter32: 17
CISCO-TCP-MIB::ciscoTcpConnInPkts.14.32.6.185.23.14.32.100.33.4184 = Counter32: 12
CISCO-TCP-MIB::ciscoTcpConnElapsed.14.32.6.185.23.14.32.100.33.4184 = Timeticks: (4345)
0:00:43.45
```

これで、このデバイスの TCP 接続をすべて確認して終了しました。

[ハングしている TCP 接続をクリアするために SNMP を使用する](#)

[手順ごとの説明](#)

SNMP を使用すると、ハングした TCP 接続をクリアできます。SNMP コマンドは、`clear tcp local <local_ip> <local_port> remote <remote_ip> <remote_port>` コマンドと同じです。1 行をクリアするために使用するオブジェクトは `tcpConnState` です。

SNMP でハングした TCP 接続をクリアするには、次のコマンドを発行します。

```
snmpset -c private nms-7206a tcpConnState.14.32.100.75.2065.172.18.86.111.23092 integer deleteTCB
```

```
TCP-MIB::tcpConnState.14.32.100.75.2065.172.18.86.111.23092 = INTEGER: deleteTCB(12)
```

注：これらのオブジェクトに使用されるインデックスは、ローカルIPアドレスの連結で、ローカルIPアドレスの14.32.100.75 (ローカルTCPポート番号2) です065、リモートIPアドレス：172.18.86.111、リモートTCPポート番号：23092。

注：「TCP 接続がハングするかを検出するために SNMP を使用する」でハングしたと判断したインデックスを正確に使用する必要があります。このコマンドが警告なく TCP 接続を切断することに注意してください。

[MIB オブジェクトの詳細情報](#)

```
.1.3.6.1.4.1.9.9.6.1.1.1.1
ciscoTcpConnInBytes OBJECT-TYPE
    -- FROM CISCO-TCP-MIB
    SYNTAX          Counter
    MAX-ACCESS      read-only
    STATUS          Current
    DESCRIPTION    "Number of bytes that have been input on this TCP
                    connection."
 ::= { ciscoTcpConnEntry 1 }

.1.3.6.1.4.1.9.9.6.1.1.1.2
ciscoTcpConnOutBytes OBJECT-TYPE
    -- FROM CISCO-TCP-MIB
    SYNTAX          Counter
    MAX-ACCESS      read-only
    STATUS          Current
    DESCRIPTION    "Number of bytes that have been output on this TCP
                    connection."
 ::= { ciscoTcpConnEntry 2 }

.1.3.6.1.4.1.9.9.6.1.1.1.3
ciscoTcpConnInPkts OBJECT-TYPE
    -- FROM CISCO-TCP-MIB
    SYNTAX          Counter
    MAX-ACCESS      read-only
    STATUS          Current
    DESCRIPTION    "Number of packets that have been input on this TCP
                    connection."
 ::= { ciscoTcpConnEntry 3 }

.1.3.6.1.4.1.9.9.6.1.1.1.4
ciscoTcpConnOutPkts OBJECT-TYPE
    -- FROM CISCO-TCP-MIB
    SYNTAX          Counter
    MAX-ACCESS      read-only
    STATUS          Current
    DESCRIPTION    "Number of packets that have been output on this TCP
                    connection."
```

```

::= { ciscoTcpConnEntry 4 }

.1.3.6.1.4.1.9.9.6.1.1.1.7
ciscoTcpConnRetransPkts OBJECT-TYPE
    -- FROM CISCO-TCP-MIB
    SYNTAX          Counter
    MAX-ACCESS      read-only
    STATUS          Current
    DESCRIPTION     "The total number of packets retransmitted due to a timeout -
                    that is, the number of TCP segments transmitted containing
                    one or more previously transmitted octets."
::= { ciscoTcpConnEntry 7 }

.1.3.6.1.4.1.9.9.6.1.1.1.9
ciscoTcpConnRto OBJECT-TYPE
    -- FROM CISCO-TCP-MIB
    SYNTAX          Integer
    MAX-ACCESS      read-only
    STATUS          Current
    DESCRIPTION     "The current value used by a TCP implementation for the
                    retransmission timeout."
::= { ciscoTcpConnEntry 9 }

.1.3.6.1.2.1.6.13.1.1
tcpConnState OBJECT-TYPE
    -- FROM RFC1213-MIB
    SYNTAX          Integer { closed(1), listen(2), synSent(3), synReceived(4),
                    established(5), finWait1(6), finWait2(7), closeWait(8), lastAck(9),
                    closing(10), timeWait(11), deleteTCB(12) }
    MAX-ACCESS      read-write
    STATUS          Mandatory
    DESCRIPTION     "The state of this TCP connection.

                    The only value which may be set by a management
                    station is deleteTCB(12). Accordingly, it is
                    appropriate for an agent to return a `badValue'
                    response if a management station attempts to set
                    this object to any other value.

                    If a management station sets this object to the
                    value deleteTCB(12), then this has the effect of
                    deleting the TCB (as defined in RFC 793) of the
                    corresponding connection on the managed node,
                    resulting in immediate termination of the
                    connection.

                    As an implementation-specific option, a RST
                    segment may be sent from the managed node to the
                    other TCP endpoint (note however that RST segments
                    are not sent reliably)."
```

```

::= { tcpConnEntry 1 }
```

[ハングした TCP 接続を検出してクリアする PERL スクリプト](#)

このリンクは、PERL スクリプトと必要な MIB モジュールのあるアーカイブ ファイルを提供します。リンクを右クリックしてファイルをシステムに保存してください。

- [fixTCPPhang.tgz](#)

アーカイブに含まれるファイルは次のとおりです。

- bin/fixTCPPhang.pl
- mibs/CISCO-SMI.my
- mibs/CISCO-TCP-MIB.my

スクリプトと MIB モジュールを抽出するには、UNIX 系のオペレーティング システムでは gzip や tar のようなユーティリティを使用します。たとえば、アーカイブ ファイルが /tmp にあると仮定してファイルを /tmp に抽出するには、次のように行います。

```
cd /tmp; gzip -dc fixTCPPhang.tgz | tar -xvf -
```

注：スクリプトの最初の行を編集して、PERLの場所を指定する必要がある場合があります。

Microsoft Windows オペレーティング システムでは、ファイルの抽出に winzip などのユーティリティを使用します。c:\tmp にファイルを抽出した場合は、スクリプトを実行するときに -m オプションを指定する必要がありません。

次のコマンドでファイルを起動します。

```
fixTCPPhang.pl -c public -C private -f nms-7206a
```

検出したハングした TCP 接続それぞれに対して、次の出力のような行が表示されます。

```
Found bad TCP connection: Local IP: 14.32.100.75 port 23 Remote IP: 172.18.100.33 port 47878:
CLEARED
```

読み書きのコミュニティ スtring が提供されて、-f オプションが指定された場合、スクリプトは接続をクリアしています。出力の最後にある CLEARED 文に注目してください。

このスクリプトは、SNMPバージョン1、2c、および3をサポートしています。SNMPバージョン3を指定する場合は、-v引数にすべての認証情報を指定する必要があります。次は SNMP v3 を使用する例です。

```
fixTCPPhang.pl -v "3 -a MD5 -u chelliot -A chelliot -l authNoPriv" -f nms-dmz-ap1200-b
```

前述の例で SNMP v3 を設定するための IOS コマンドは次のとおりです。

```
snmp-server group chelliot-group v3 auth write v1default
snmp-server user chelliot chelliot-group v3 auth md5 chelliot
```

注：このテストで使用されているNET-SNMPのWindowsバージョンにバグがあるようです。このバグによって、SHA 認証は正しく動作できません。

このスクリプトで使用できるオプションは他にいくつかあります。NET-SNMP コマンドライン ユーティリティの場所や MIB モジュールの場所が /tmp/mibs にない場合に含めるスクリプト オプションがいくつかあります。また、それらのオプションの次のような要約も表示できます。

fixTCPPhang.pl

```
fixTCPPhang.pl [-dfhV -c <read_community> -C <write_community> -m <mib_directory>
                -p <command_path> -t <timeout> -v <snmp_version>] <device>
```

Version 1.2

Detect hung TCP connections on <device>, optionally clearing them.

Options:

- c Specify read community string. Defaults to public.
- C Specify the readwrite community string. No default.
Must be supplied for the script to clear hung connections.
- d Turn on debug mode.
- f Fix or clear any hung TCP connections found.
- h Print this message.
- m Specify the directory to find CISCO-SMI.my and CISCO-TCP-MIB.my.
Defaults to /tmp/mibs.
- p Where to find the net-snmp utilities.
Optional if the utilities are in the path.
- t SNMP Timeout value. Defaults to 5 sec.
- v Specify SNMP version to use: One of 1, 2c, or 3.
If 3 is specified then this option must include all of the authentication information for SNMPv3. For example:
"3 -a MD5 -u chelliot -A chelliot -l authNoPriv"
Note: NET-SNMP seems to have a bug with SHA authentication on Windows.
See the NET-SNMP documentation for more information.
Defaults to SNMP version 1.
- V Print version number.

関連情報

- [テクニカルサポート - Cisco Systems](#)