ベースライン プロセスのベスト プラクティスのホワイト ペーパー

内容

はじめに

この文書では、可用性の高いネットワークを構築するためのベースラインのコンセプトと手順を説明します。これには、成功の評価を行うためのネットワーク ベースラインとしきい値設定に関する重要な成功要因が含まれています。さらに、シスコのハイ アベイラビリティ サービス（HAS）チームによって明らかにされた最適な方法のガイドラインに基づき、ベースラインとしきい値プロセスおよび実装について詳細に説明します。

この文書では、ベースラインのプロセスを手順を追って実行します。現在の Network Management System（NMS; ネットワーク管理システム）の製品によっては、このプロセスを自動化できますが、自動ツール、手動ツールのいずれを使用しても、ベースラインのプロセス自体は同じです。次のNMSの製品を使用すると、独自のネットワーク環境のデフォルトのしきい値の設定を調整します。有効、正確にプロセスをインテリジェントにこれらのしきい値を選択して使

用することが重要です。

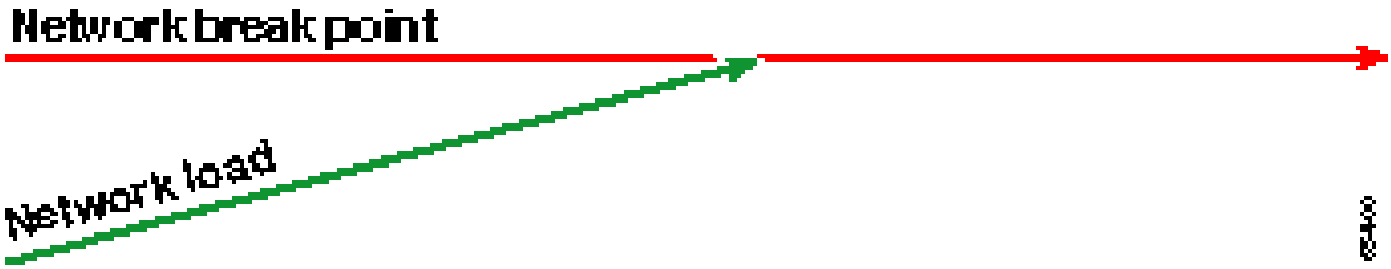
ベースライン

ベースラインとは何か

ベースラインとは、ネットワークを定期的に調査してネットワークが確実に設計どおりに動作するようにするためのプロセスです。ベースラインは、特定時点のネットワークの状態を詳細に言及する一つのレポートにとどまりません。ベースラインプロセスに従って、次の情報を入手できます:

- ハードウェアおよびソフトウェアの健全性、有用な情報を取得します
- 現在のネットワーク使用のリソースを決定します
- ネットワーク アラームのしきい値に関する正確な決定を行います
- 現在のネットワーク問題を特定します
- 将来の問題を予測できる。

ベースラインを表示する別の例を次の図に説明されます。



ネットワークのブレーク ポイントを示す赤いラインは、ネットワークがブレークするポイントを示します。このポイントは、ハードウェアおよびソフトウェアの動作方法に関する情報に基づいて決まります。ネットワーク負荷を意味する緑のラインは、新しいアプリケーションの追加やその他の要因により、ネットワークの負荷が自然に増えていく状態を示します。

ベースラインの目標は、次の事項を確認することです。

- ネットワークが緑のラインのどの位置にいるのか。
- 速度ネットワーク負荷が増大している
- One canどの時点で、を通過することを予測します

ベースラインを定期的に実行することで、障害がそれに事前に実行され、対応すると現在の状態を確認し、推定数を見積もることができます。さらに、ネットワークのアップグレードについて、予算額をいつ、どこで、どう使用するかを、情報に基づいて決定できます。

なぜベースラインなのか

ベースラインプロセスがネットワークの重要なリソース制限問題の識別して適切に計画できます。これらの問題は、コントロール プレーン リソースまたはデータ プレーン リソースとして説明できます。コントロール プレーンのリソースは、デバイス内の特定のプラットフォームおよびモジュールに固有で、一部の問題によっても影響する方法:

- データの使用状況
- 有効な機能
- ネットワーク設計

コントロール プレーンのリソースはパラメータをなどの:

- CPU Utilization
- メモリ使用率
- バッファ使用率

データ プレーンのリソースは、タイプおよびトラフィックだけが許可され、リンク利用率とバックプレーン使用率が含まれます。重要な領域のベースライン リソースの使用によって、重大なパフォーマンスの問題やより深刻なネットワークメルトダウンを回避できます。

音声やビデオなどの遅延の影響を受けやすいアプリケーションの導入により、ベースラインはより重要になっています。一般的な Transmission Control Protocol/Internet Protocol（TCP/IP）アプリケーションは、一定の遅延を許容できます。一方、音声とビデオのアプリケーションの場合は、User Datagram Protocol（UDP）に基づいており、再転送やネットワークの輻輳を許容しません。

アプリケーションの新しい混合、ベースラインが原因でコントロール プレーンとデータ プレーンの両方のリソース使用率の問題を特定し、プロアクティブな継続的成功を保証する変更およびアップグレードを計画できます。

データは長期にわたっています。最近まで、ネットワークを稼動することは、多少の工ラーも比較的許容されていました。Voice over IP（VoIP）などの遅延の影響を受けやすいアプリケーションが急速に受け入れられるにつれて、ネットワークの稼動は、より困難になるとともに高い精度が要求されています。より正確にするとネットワーク管理者に把握することが重要ネットワークがどのように動作するかネットワークを管理するために、概念が強固な基盤を提供します。そのためには、ベースラインと呼ばれるプロセスを導入する必要があります。

ベースライン目標

ベースラインの目標は、次の事項を実行することです。

1. ネットワーク デバイスの現在の状態を判別する。
2. 標準性能のガイドラインとその状態を比較します
3. ネットワーク装置の状況がガイドラインを超えたら、警告するようにしきい値を設定する。

データ分析に要する時間と大量のデータが原因でプロセスを確認すると、基準の範囲を、より簡単にに限定する必要があります。ネットワークのコア部分から開始するのが、最も合理的で、通常、最も有益です。ネットワークのこの部分は、通常、最も小さく、しかも最高の安定性が求められるためです。

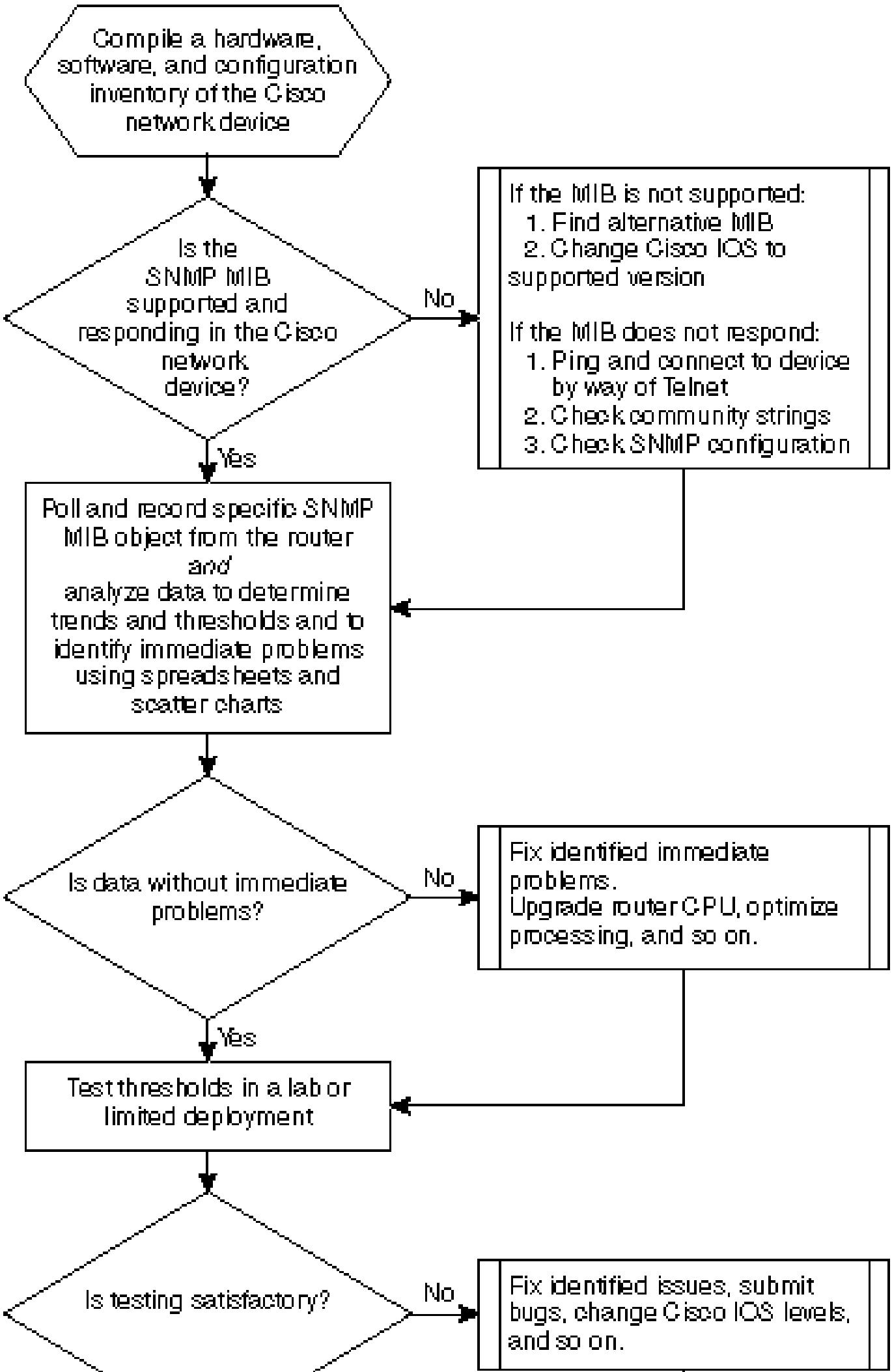
ここでは、簡単にするために、1つの非常に重要なSimple Network Management Protocol(SNMP)Management Information Base(SNMP MIB)cpmCPUTotal5minのベースラインを設定する方法について説明します。cpmCPUTotal5minは、Ciscoルータの中央処理装置(CPU)の5分間の減衰平均で、コントロールプレーンのパフォーマンスインジケータです。このベースラインは、Cisco 7000 シリーズのルータで実行します。

プロセスを学習し終えたら、大規模な SNMP データベース内の任意のデータに適用できます。次のような SNMP データベースは、ほとんどの Cisco 装置に用意されています。

- 統合サービス デジタル網（ISDN）の使用状況
- 非同期転送モード（ATM）セル損失
- システム メモリの空き容量

コア ベースライン フローチャート

次のフローチャートでは、コア ベースライン プロセスの基本的なステップを示しています。これらのステップを実行するときに使える製品やツールも用意されていますが、柔軟性や使いやすさの点で差があります。ベースラインを使用して、ネットワーク管理システム（NMS ）ツールを使用することを計画している場合も、プロセスを調査し、ネットワークが実際にする方法を理解するための適切な課題です。ほとんどのツールは本質的には同じことをするため、このプロセスにより、いくつかの NMS ツールの動作方法に関する疑問が解決する場合があります。


Compile a hardware, software, and configuration inventory of the Cisco network device
Is the SNMP MIB supported and responding in the Cisco network device?
No
If the MIB is not supported:
1. Find alternative MIB
2. Change Cisco IOS to supported version
If the MIB does not respond:
1. Ping and connect to device by way of Telnet
2. Check community strings
3. Check SNMP configuration
Yes
Poll and record specific SNMP MIB object from the router
and
analyze data to determine trends and thresholds and to identify immediate problems using spreadsheets and scatter charts
Is data without immediate problems?
No
Fix identified immediate problems.
Upgrade router CPU, optimize processing, and so on.
Yes
Test thresholds in a lab or limited deployment
Is testing satisfactory?
No
Fix identified issues, submit bugs, change Cisco IOS levels, and so on.

ベースライン手順

ステップ1: ハードウェア、ソフトウェアおよび設定インベントリのコンパイル

ハードウェア、ソフトウェア、設定のインベントリをいくつかの理由で編集することは非常に重要です。まず、Cisco SNMP MIBは、実行中のCisco IOS Releaseに、場合によっては、仕様です。MIB オブジェクトによっては、新しいものと交換されたり、また完全に破棄されます。データ収集後には、ハードウェアのインベントリが最も重要です。これは、最初のベースライン実行後に、Cisco 装置上の CPU のタイプ、メモリ容量などに基づいてしきい値を設定する場合が多いためです。現在の設定を確認するには、設定インベントリも重要です。ベースラインの後にデバイス設定を変更して、バッファを調整する場合などです。

Cisco ネットワークの場合、ベースラインのステップ 1 の部分を最も効率的に実行するには、CiscoWorks2000 Resource Manager Essentials（Essentials）を使用します。このソフトウェアがネットワークに正常にインストールされている場合は、コメントは、データベース内のすべてのデバイスの現在のインベントリが必要です。問題の有無を確認するには、そのインベントリを参照するだけで済みます。

次の表は、Essentials からエクスポートされた Cisco Router Class ソフトウェアのインベントリレポートを Microsoft Excel で編集した例です。このインベントリから、SNMP MIBのデータを使用して、オブジェクトID（OID）が12.0xおよび12.1xのCisco IOSリリースであることに注意してください。

Device Name	ルータタイプ	バージョン	[Software Version]
field-2500a.embu-mlab.cisco.com	Cisco 2511	M	12.1(1)
qdm-7200.embu-mlab.cisco.com	Cisco 7204	B	12.1(1)E
voip-3640.embu-mlab.cisco.com	Cisco 3640	0x00	12.0(3c)
WAN 1700a.embu mlab.cisco.com	Cisco 1720	0x101	12.1(4)
WAN 2500a.embu mlab.cisco.com	Cisco 2514	起	12.0(1)
WAN 3600a.embu mlab.cisco.com	Cisco 3640	0x00	12.1(3)
wan-7200a.embu-mlab.cisco.com	Cisco 7204	B	12.1(1)E
172.16.71.80	Cisco 7204	B	12.0(5T)

概要がネットワークにインストールされていない場合は、IOSバージョンを検索します (UNIXワークステーションからUNIXのコマンドライン ツールでsnmpwalkを使用できます。これを次の例で示します。このコマンドがどのように機能するかわからない場合は、UNIX プロンプトで snmpwalk を入力して詳細を確認します。どの MIB OID をベースラインにするかを選択する場合、IOS バージョンが重要になります。これは MIB オブジェクトが IOS に依存しているためです。また、しきい値がCPUのバッファ用などであるものについてことをルータが認識することにより、次を決定できることに注意してください。

```
nsahpov6% snmpwalk -v1 -c private 172.16.71.80 system
system.sysDescr.0 : DISPLAY STRING- (ascii): Cisco Internetwork Operating System Software
IOS (tm) 7200 Software (C7200-JS-M), Version 12.0(5)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Fri 23-Jul-2001 23:02 by kpma
system.sysObjectID.0 : OBJECT IDENTIFIER:
.iso.org.dod.internet.private.enterprises.cisco.ciscoProducts.cisco7204
```

ステップ2: SNMP MIB がルータでサポートされることの確認

ベースラインにポーリングするデバイスのインベントリがあるため、ポーリングする特定のOIDを選択できます。これは目的のデータで実際に存在し、事前に確認する多くの不満を保存します。cpmCPUTotal5min MIB オブジェクトは、CISCO-PROCESS-MIB にあります。

ポーリングする OID を探すには、Cisco の CCO ウェブ サイトにある変換テーブルが必要です。Web ブラウザからこの Web サイトにアクセスするには、Cisco MIB ページにアクセスして、OIDのリンクをクリックします。

FTP サーバからこの Web サイトにアクセスするには、ftp://ftp.cisco.com/pub/mibs/oid/ と入力します。このサイトから、OID番号解釈およびソートされた特定のMIBをダウンロードできます。

次の例に、CISCO-PROCESS-MIB.oid テーブル一部を示します。この例は、cpmCPUTotal5min MIB の OID が .1.3.6.1.4.1.9.9.109.1.1.1.1.5 であることを示しています。

注：OIDの先頭に必ず「。」を追加してください。そうしないと、ポーリングしようとすると工ラーが発生します。また、OID をインスタンスにするには、OID の最後に「.1」を追加する必要があります。これによって、探している OID のインスタンスが装置に伝えられます。場合によっては、ルータが複数のCPUがある場合、OIDのデータの特定のタイプの複数のインスタンスがあります。

<#root>

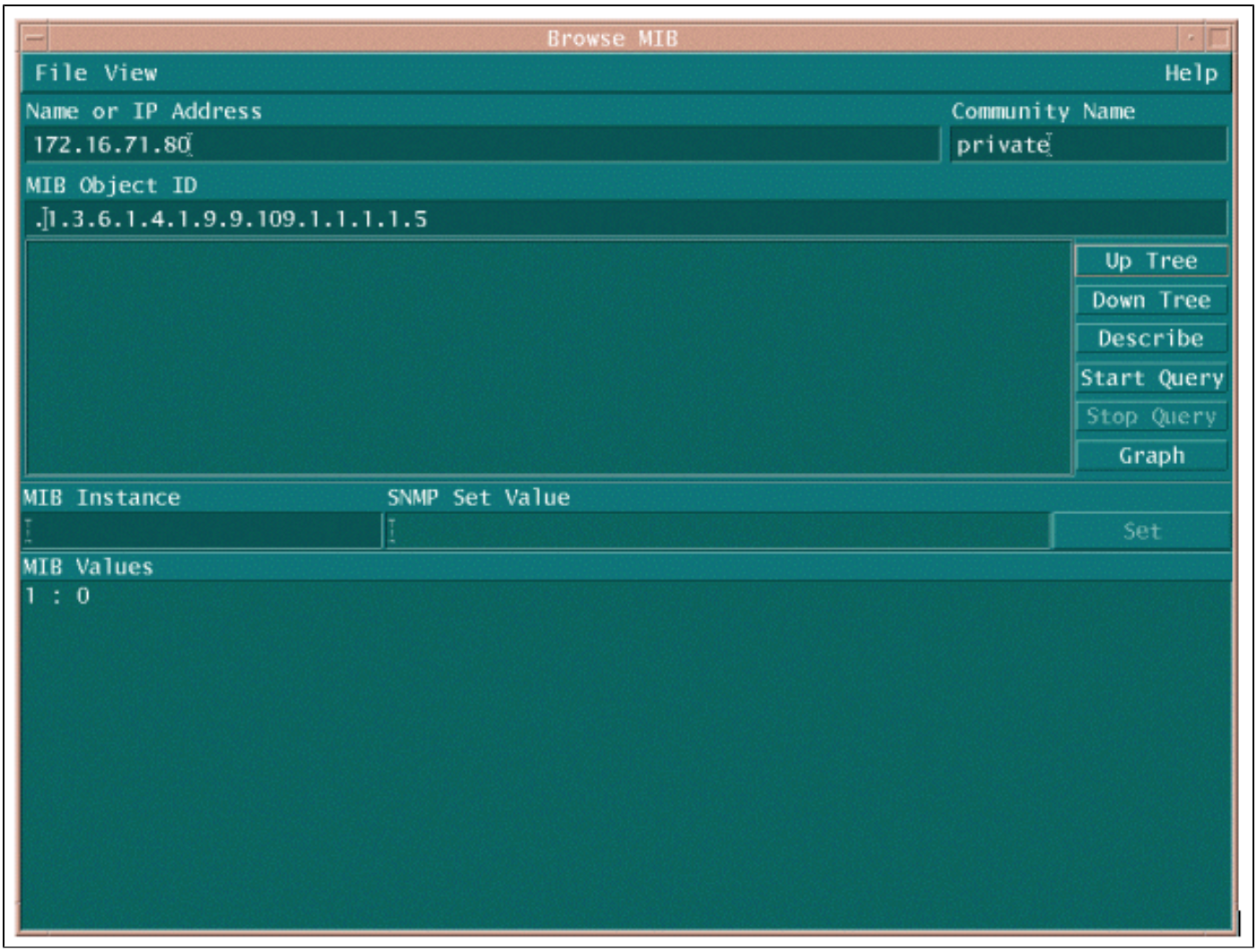
```
ftp://ftp.cisco.com/pub/mibs/oid/CISCO-PROCESS-MIB.oid
### THIS FILE WAS GENERATED BY MIB2SCHEMA
"org" "1.3"
"dod" "1.3.6"
"internet" "1.3.6.1"
"directory" "1.3.6.1.1"
"mgmt" "1.3.6.1.2"
"experimental" "1.3.6.1.3"
```

```
"private" "1.3.6.1.4"
"enterprises" "1.3.6.1.4.1"
"cisco" "1.3.6.1.4.1.9"
"ciscoMgmt" "1.3.6.1.4.1.9.9"
"ciscoProcessMIB" "1.3.6.1.4.1.9.9.109"
"ciscoProcessMIBObjects" "1.3.6.1.4.1.9.9.109.1"
"ciscoProcessMIBNotifications" "1.3.6.1.4.1.9.9.109.2"
"ciscoProcessMIBConformance" "1.3.6.1.4.1.9.9.109.3"
"cpmCPU" "1.3.6.1.4.1.9.9.109.1.1"
"cpmProcess" "1.3.6.1.4.1.9.9.109.1.2"
"cpmCPUTotalTable" "1.3.6.1.4.1.9.9.109.1.1.1"
"cpmCPUTotalEntry" "1.3.6.1.4.1.9.9.109.1.1.1.1"
"cpmCPUTotalIndex" "1.3.6.1.4.1.9.9.109.1.1.1.1.1"
"cpmCPUTotalPhysicalIndex" "1.3.6.1.4.1.9.9.109.1.1.1.1.2"
"cpmCPUTotal5sec" "1.3.6.1.4.1.9.9.109.1.1.1.1.3"
"cpmCPUTotal1min" "1.3.6.1.4.1.9.9.109.1.1.1.1.4"

"cpmCPUTotal5min" "1.3.6.1.4.1.9.9.109.1.1.1.1.5"
```

MIB OID が使用可能であり、動作していることを確認するために、MIB OID をポーリングする 場合は 2 つの一般的な方法があります。時間のポーリングのない浪費し、空のデータベースに終らない何かようにバルク データ収集を開始する前に実行することを推奨します。MIB OID をポーリングする 1 つの方法に、HP OpenView Network Node Manager（NNM）、または CiscoWorks Windows などの NMS プラットフォームから MIB ウォーカを使用し、チェックする OID を入力する方法があります。

次に、HP OpenView SNMP MIB ウォーカの例を示します。



MIB OIDをポーリングするもう一つの簡単な方法は、次の例に示すように、UNIXコマンドでsnmpwalkを使用します。

```
nsahpov6% cd /opt/OV/bin
nsahpov6% snmpwalk -v1 -c private 172.16.71.80 .1.3.6.1.4.1.9.9.109.1.1.1.1.5.1

cisco.ciscoMgmt.ciscoProcessMIB.ciscoProcessMIBObjects.cpmCPU.cpmCPUTotalTable.cpmCPUTotalEntry.cpmCPUT
```

どちらの例では、MIBは、ポーリング サイクルにCPUが0の使用率が平均したことを意味する0を返しました。正しいデータで応答するデバイスに問題がある場合は、デバイスのpingやTelnet経由でデバイスにアクセスしてください。それでも問題がある場合は、SNMP設定とSNMPコミュニティ ストリングをオンにします。問題を解決するには、別の MIB または IOS の他のバージョンを探す必要がある場合があります。

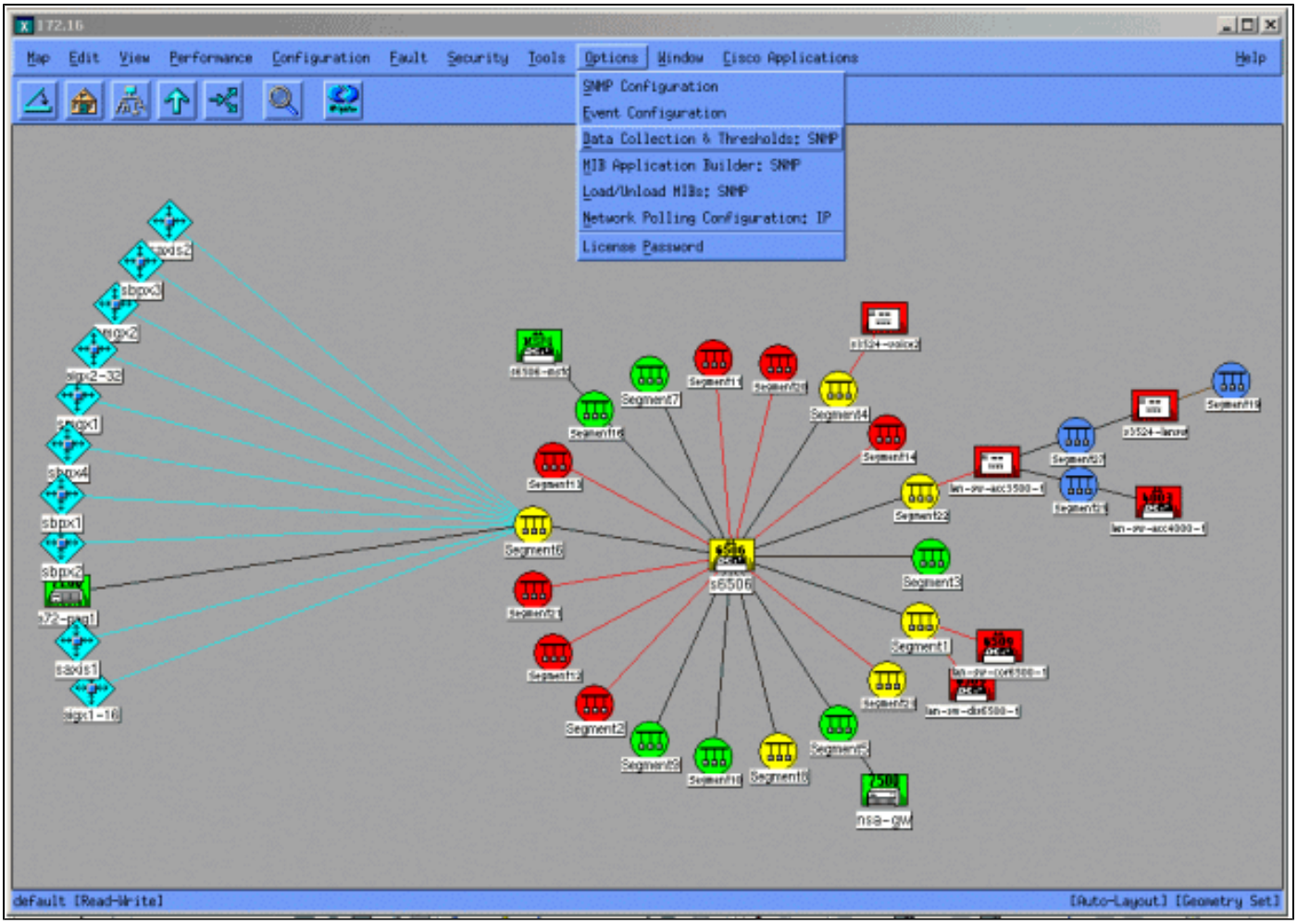
ステップ3: ポーリングおよびレコードルータからの特定のSNMP MIB オブジェクト

MIB オブジェクトをポーリングして、その出力結果を記録する方法がいくつかあります。市販製品、シェアウェアの製品、スクリプトおよびベンダー ツールを使用できます。すべてのフロント

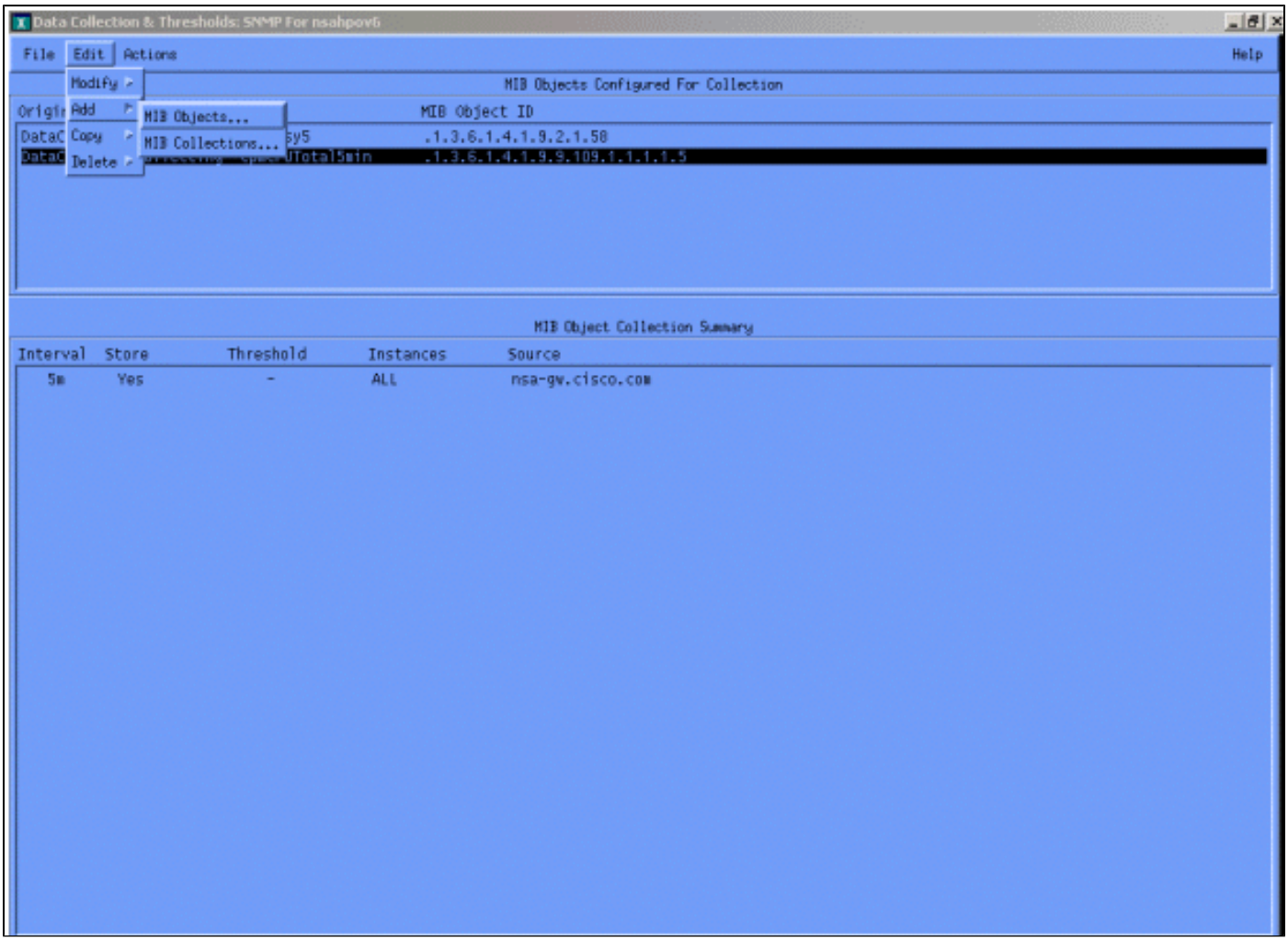
エンド ツールはSNMP GET情報を取得するプロセスを使用します。主に、構成の柔軟性とデータをデータベースに記録する方法が異なります。また、これらのさまざまな方法がどのように機能するかプロセッサMIBの概要。

確認のため、OIDはルータでサポートされているをポーリングするための、頻度とそれを記録する方法を決定する必要があります。シスコはCPU MIBが5分間隔でポーリングされることを推奨します。低い間隔は、ネットワークまたはデバイスの負荷が増大し、MIB値はとにかく5分平均であるため、多くの場合、平均値をポーリングすると便利です。ネットワーク内の少なくとも二つの週ビジネス循環を分析できるようにベースラインのポーリングに2週間の期間であることが、一般に推奨されます。

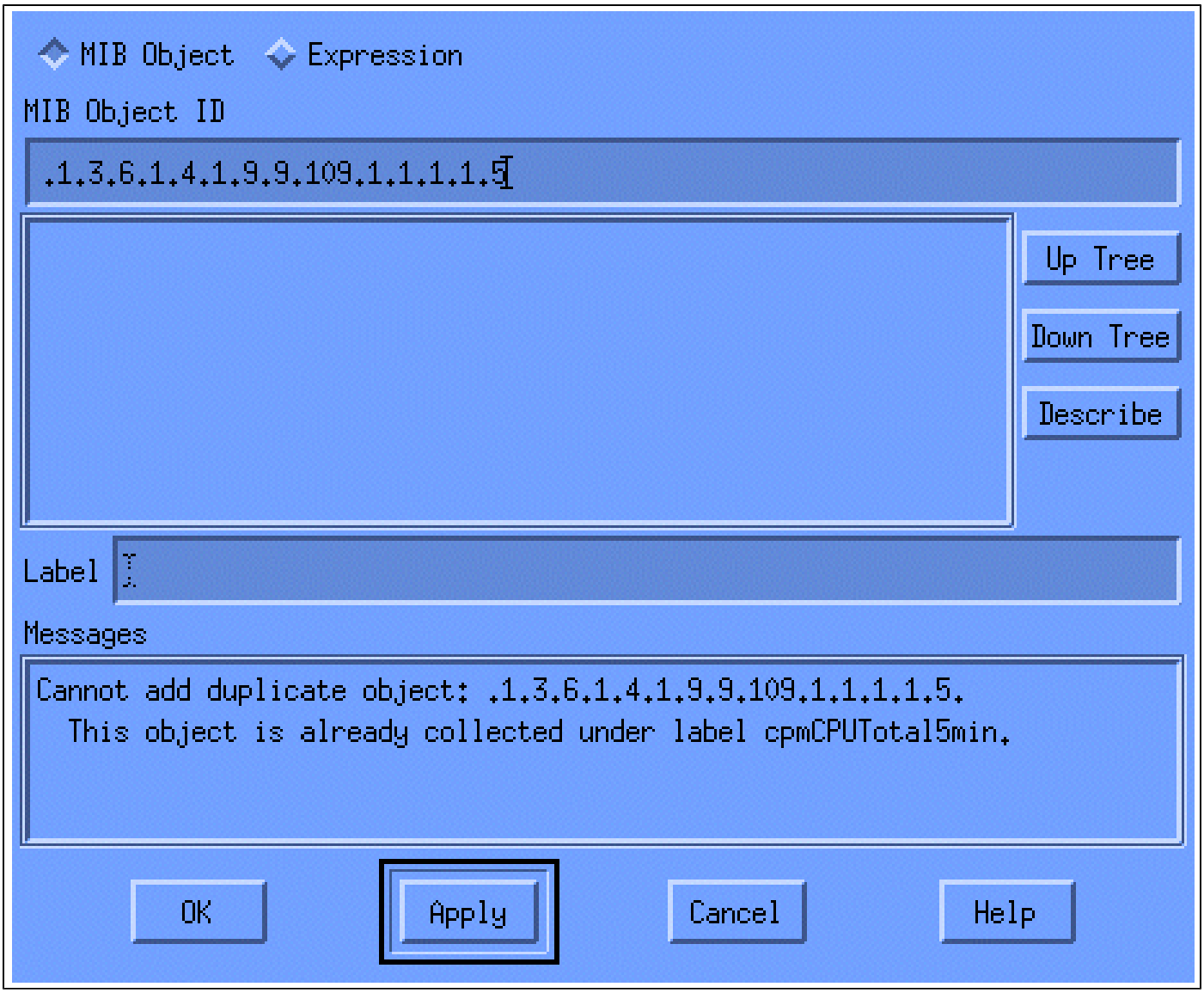
次の画面では、HP OpenView Network Node Manager version 6.1 によって MIB オブジェクトを追加する方法を示しています。メイン画面から、Options > Data Collection & Thresholdsの順に選択してください。



[Edit > Add > MIB Objectsの順に選択してください。

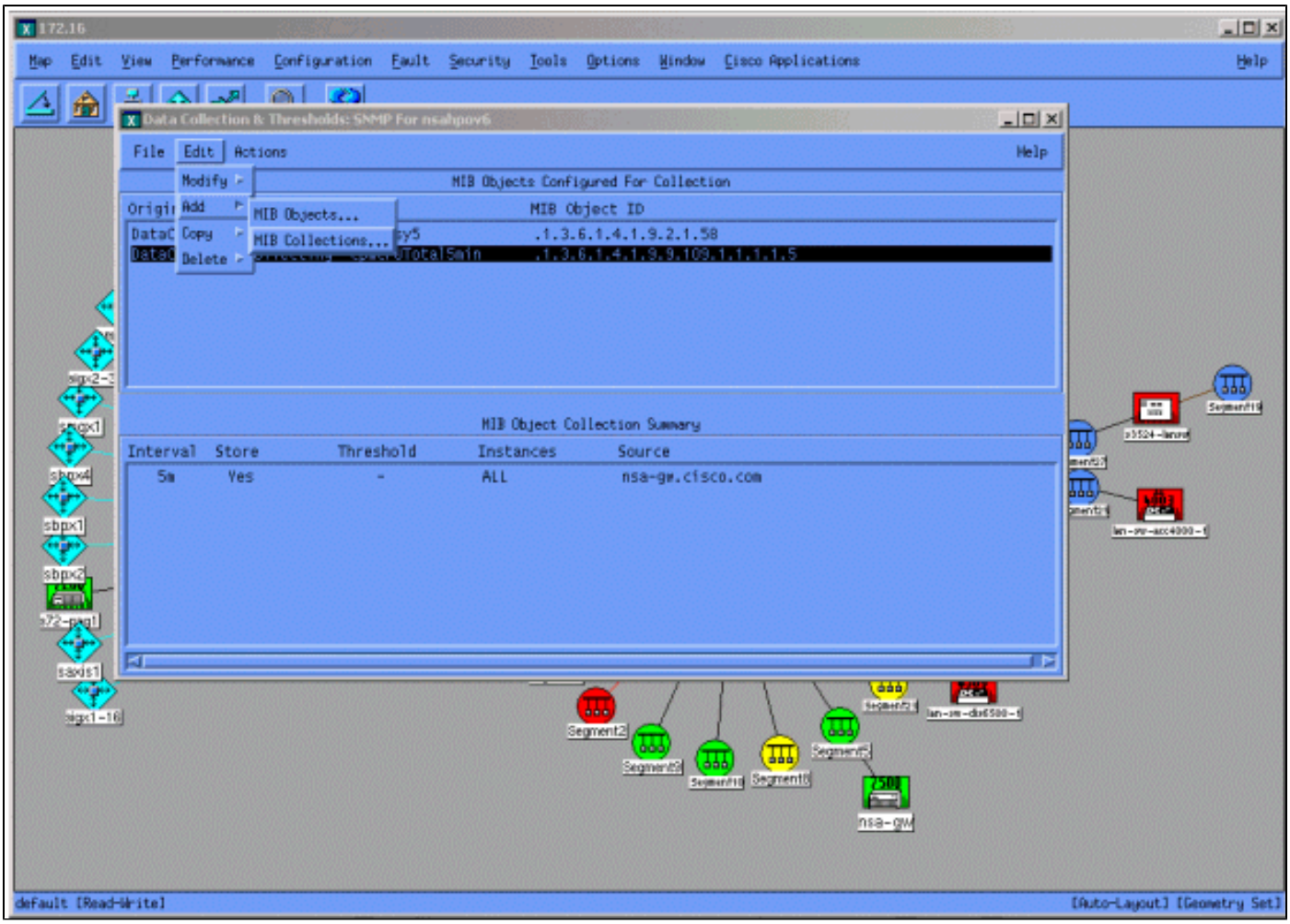


メニューから、OIDストリングを入力し、[Apply]をクリックします。これで、HP OpenView プラットフォームに MIB オブジェクトが入力されたので、ポーリングすることができます。



次に、この OID について、ポーリングするルータを HP OpenView に指示する必要があります。

[Data Collection]メニューの[Edit > Add > MIB Collectionsの順に選択してください。

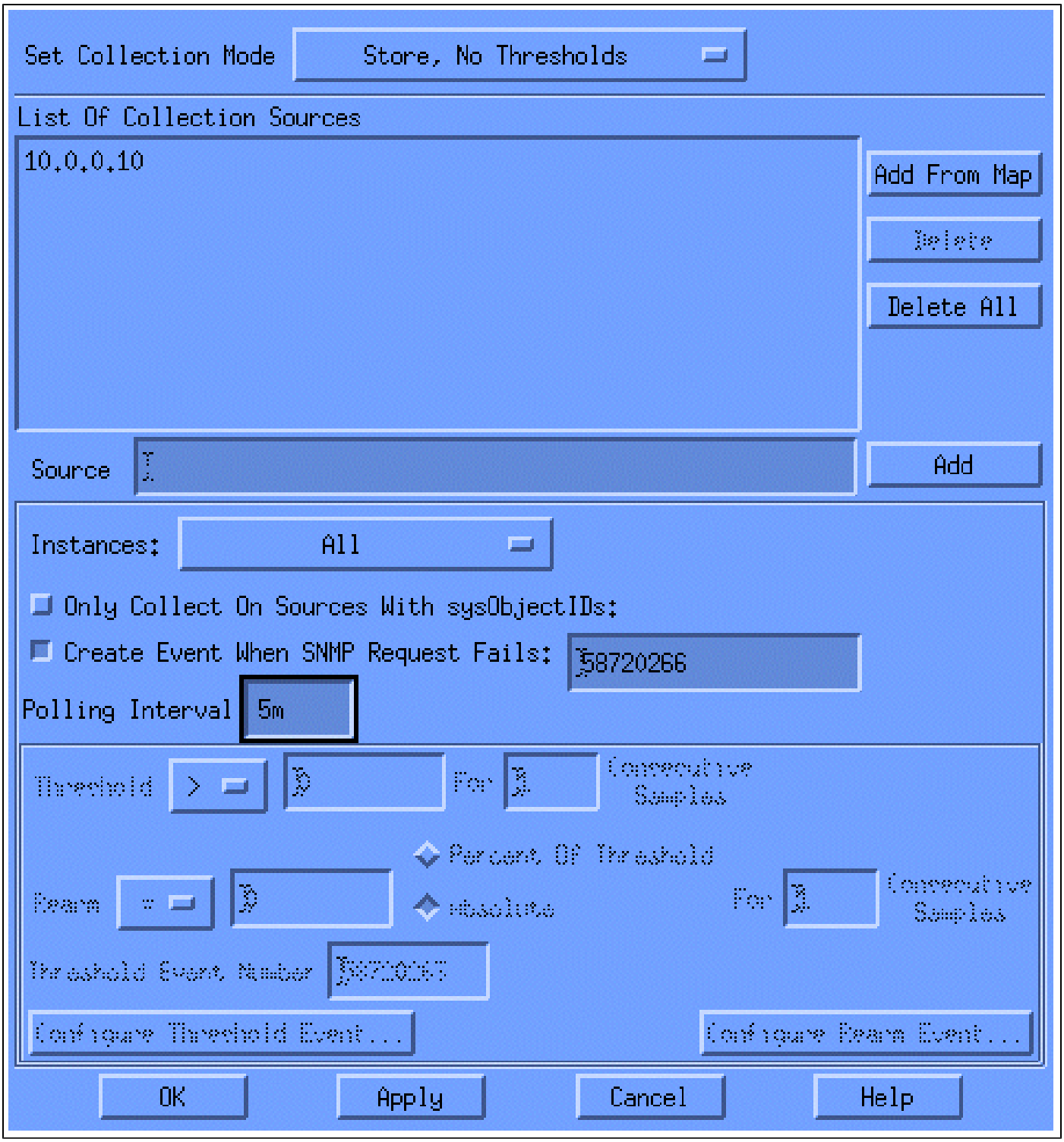


[Source]フィールドで、ポーリングされるルータのシステム（DNS ）の名前またはIPアドレスを指定するドメインを入力します。

[Set Collection Mode] リストから [Store, No Thresholds] を選択します。

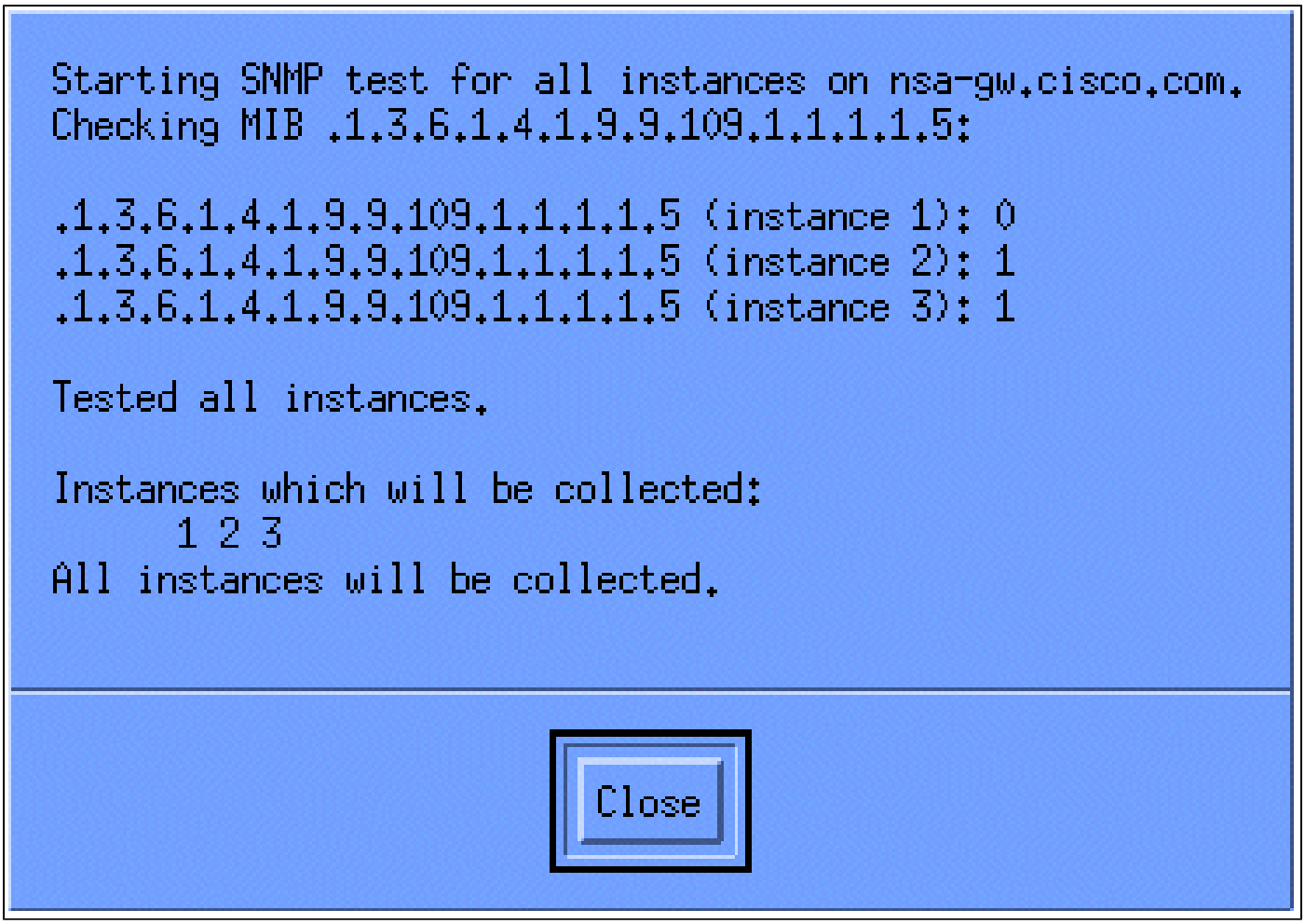
[Polling Interval] を [5m]（5 分間隔）に設定します。

[APPLY] をクリックします。



変更を有効にするため、[File] > [Save] を選択する必要があります。

収集が正しく設定されていることを確認するには、ルータのコレクションのサマリー行を強調表示し、[Actions > Test SNMPの順に選択してください。この結果、コミュニティ ストリングが正確であるかどうか、および OID の全インスタンスがポーリング対象となっているかどうかをチェックできます。



[Close]をクリックして、収集が1週間動作させます。毎週期間の終了時に、分析データを取得します。

データを分析しやすくするには、ASCII ファイルにデータをダンプして、Microsoft Excel などの表計算ツールにインポートします。HP OpenView NNM を使用して同じことを実行する場合、コマンドライン ツール snmpColDump を使用できます。設定された各収集は /var/opt/OV/share/databases/snmpCollect/のディレクトリにファイルを書き込みます。

"次のコマンドで、テストファイル内ASCIIファイルにデータの取得:

<#root>

```
snmpColDump /var/opt/OV/share/databases/snmpCollect/cpmCPUTotal5min.1 >
```

testfile

注：cpmCPUTotal5min.1は、OIDポーリングの開始時にHP OpenView NNMが作成したデータベースファイルです。

作成されたテスト ファイルは、次の例のように表示されます。

```
03/01/2001 14:09:10 nsa-gw.cisco.com 1
03/01/2001 14:14:10 nsa-gw.cisco.com 1
03/01/2001 14:19:10 nsa-gw.cisco.com 1
03/01/2001 14:24:10 nsa-gw.cisco.com 1
03/01/2001 14:29:10 nsa-gw.cisco.com 1
03/01/2001 14:34:10 nsa-gw.cisco.com 1
03/01/2001 14:39:10 nsa-gw.cisco.com 1
03/01/2001 14:44:10 nsa-gw.cisco.com 1
03/01/2001 14:49:10 nsa-gw.cisco.com 1
03/01/2001 14:54:10 nsa-gw.cisco.com 1
03/01/2001 14:59:10 nsa-gw.cisco.com 1
03/……
```

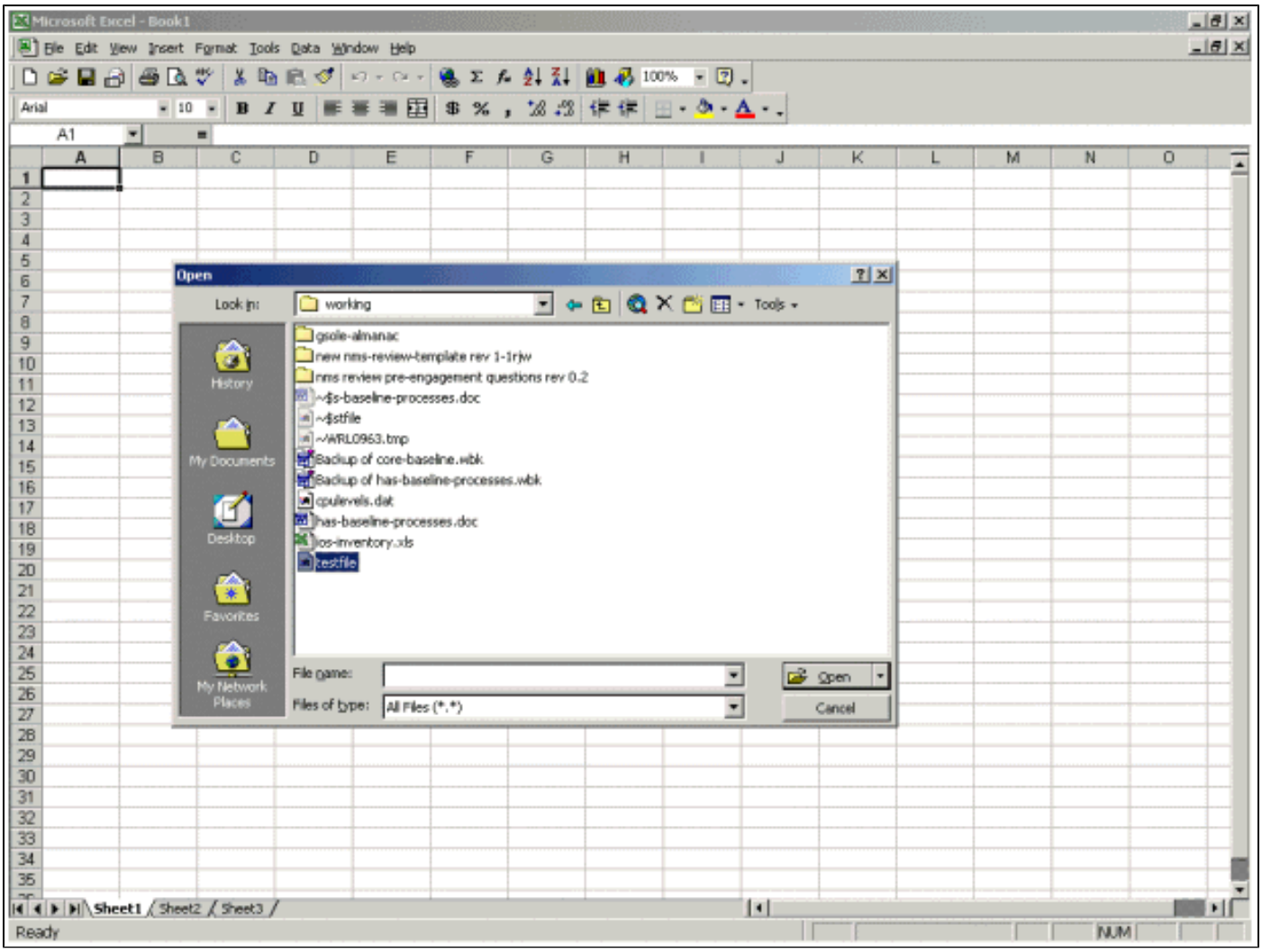
テスト ファイルの出力結果が UNIX 端末に表示されたら、File Transfer Protocol（FTP）を使用して各自の PC に転送できます。

自分のスクリプトを使用してもデータを収集できます。そのためには、5 分ごとに CPU OID に対して snmpget コマンドを実行して、結果を .csv ファイルにダンプします。

ステップ4: しきい値を判別するためにデータを分析する

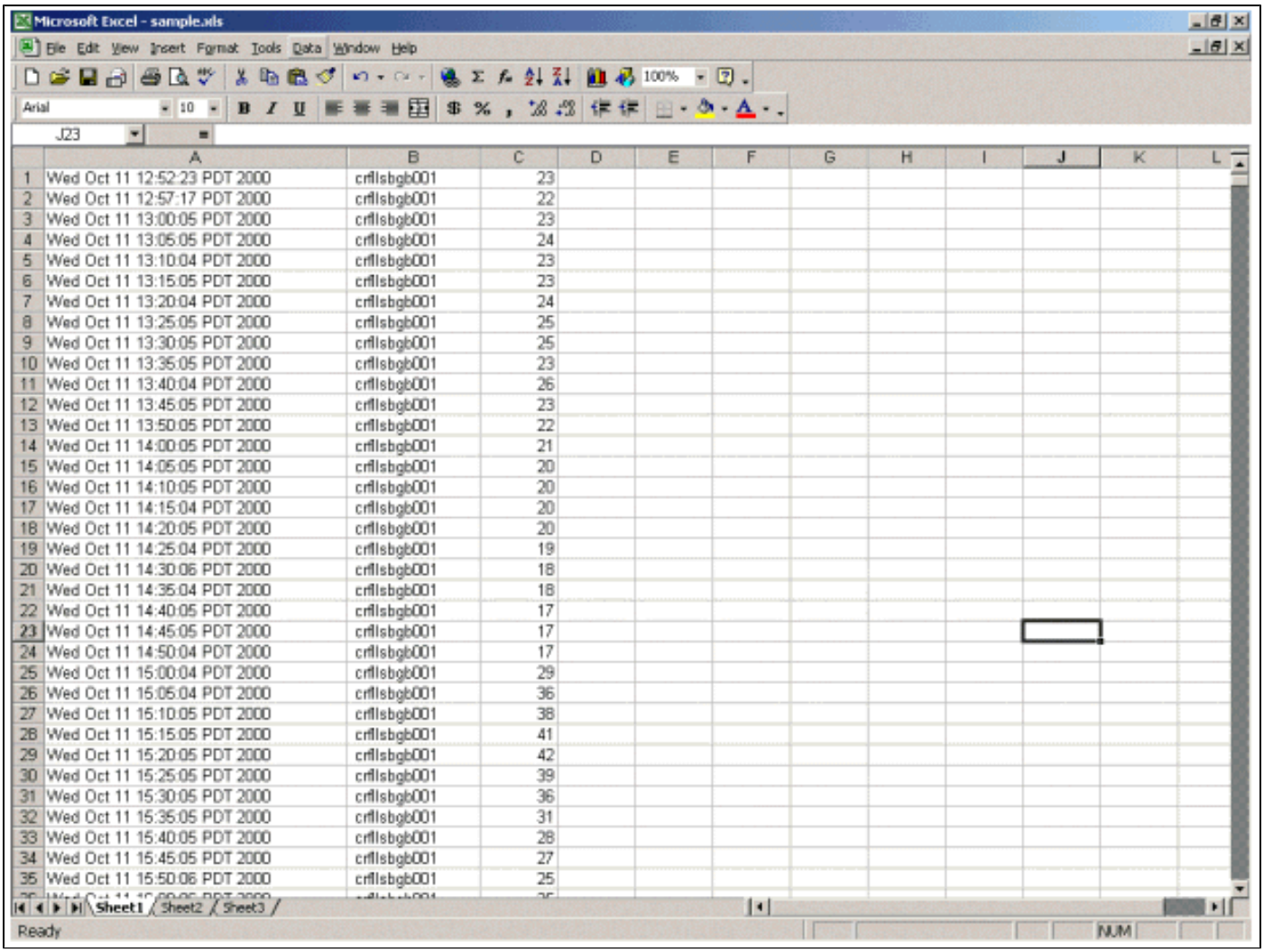
あるデータがあるため、分析できます。この段階のベースラインでは、しきい値を設定します。しきい値を設定することで、パフォーマンスや障害を正確に測定でき、しかもしきい値のモニタリングをオンにしておくと、アラームが発生し過ぎません。最も簡単な設定方法の 1 つは、データを Microsoft Excel などの表計算にインポートしてから、散布図にプロットする方法です。この方法を使用すると、特定のしきい値についてある装置を監視している場合に、その装置が例外アラートを生成する回数を簡単に確認できます。ベースラインをnotしきい値にすることはお勧めできませんこれが選択したしきい値を超過したデバイスからアラート ストームを作成する場合があります。

テスト ファイルをExcelスプレッドシートにインポートし、Excelを開き、データ ファイルを選択する[File] > [Open]を選択し。



次に、ファイルのインポート中に、Excel アプリケーションから確認を求められます。

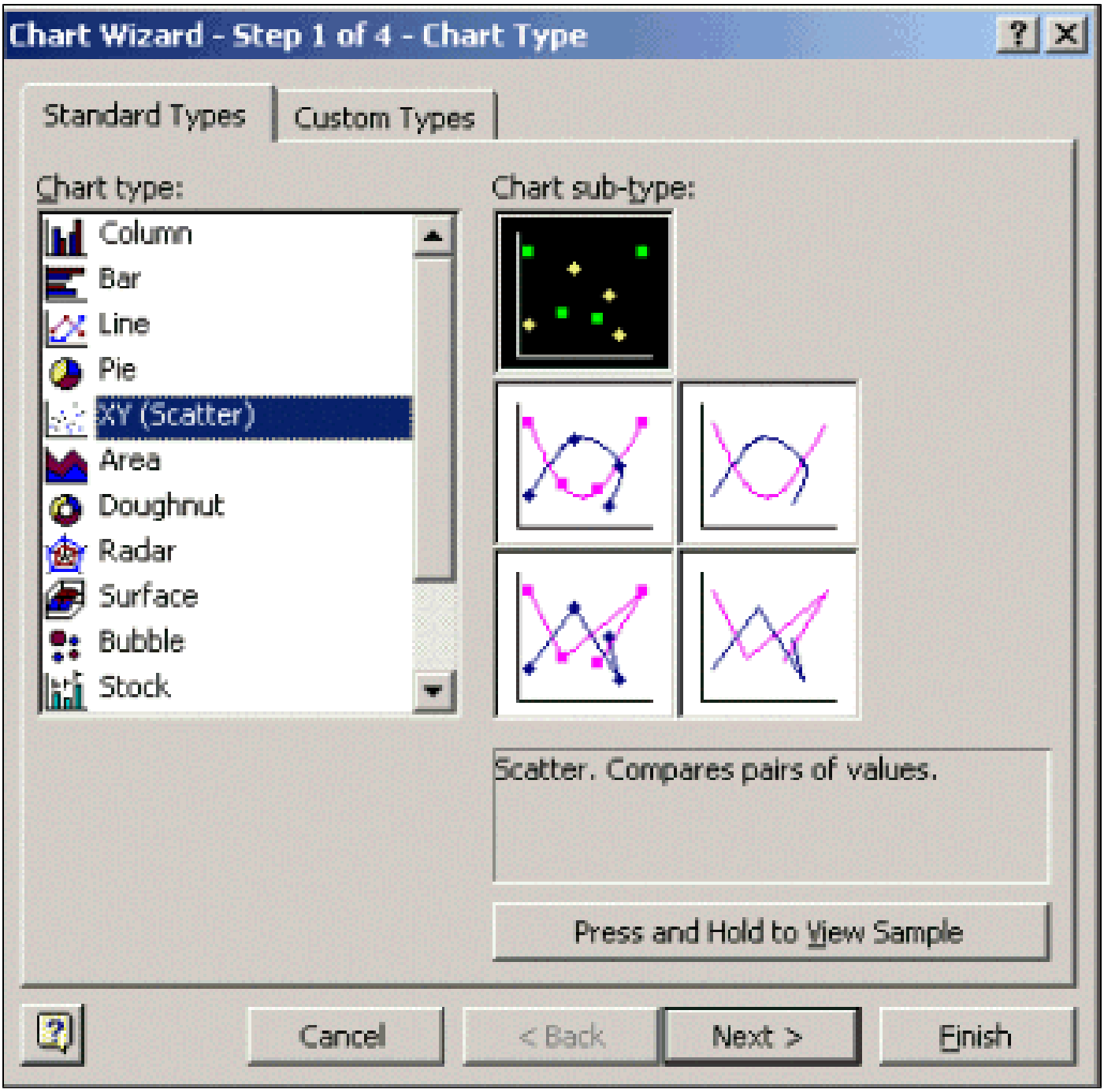
完了すると、インポートされたファイルは次の画面のように表示されます。



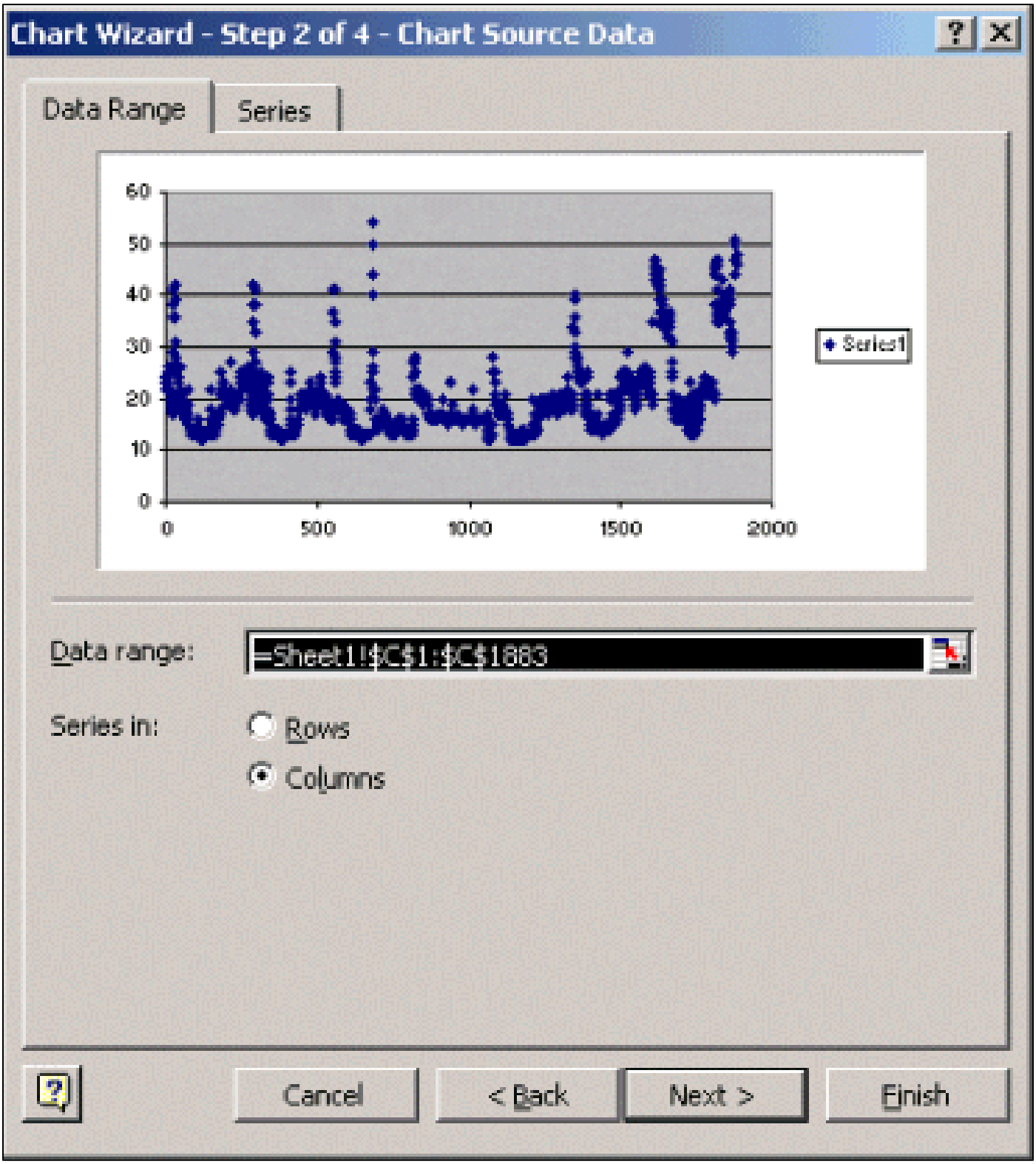
散布図はさまざまなしきい値設定がネットワークでどのように動作するかをより簡単にイメージが許可されます。

散布図を作成するには、インポートしたファイルの列 C を強調表示させ、[Chart Wizard] アイコンをクリックします。次に、Chart Wizard のステップに従って、散布図を作成します。

次に示すチャート ウィザード ステップ1では、Standard Types]タブを選択し、XY（散布図）形式を選択します。次に [Next] をクリックします。



次に示すチャート ウィザード ステップ2では、データ範囲タブを選択し、日付範囲と[Columns]オプションを選択します。[Next] をクリックします。



次に示すチャート ウィザード ステップ3で、グラフ タイトルとXとY軸の値を入力し、[Next]をクリックします。

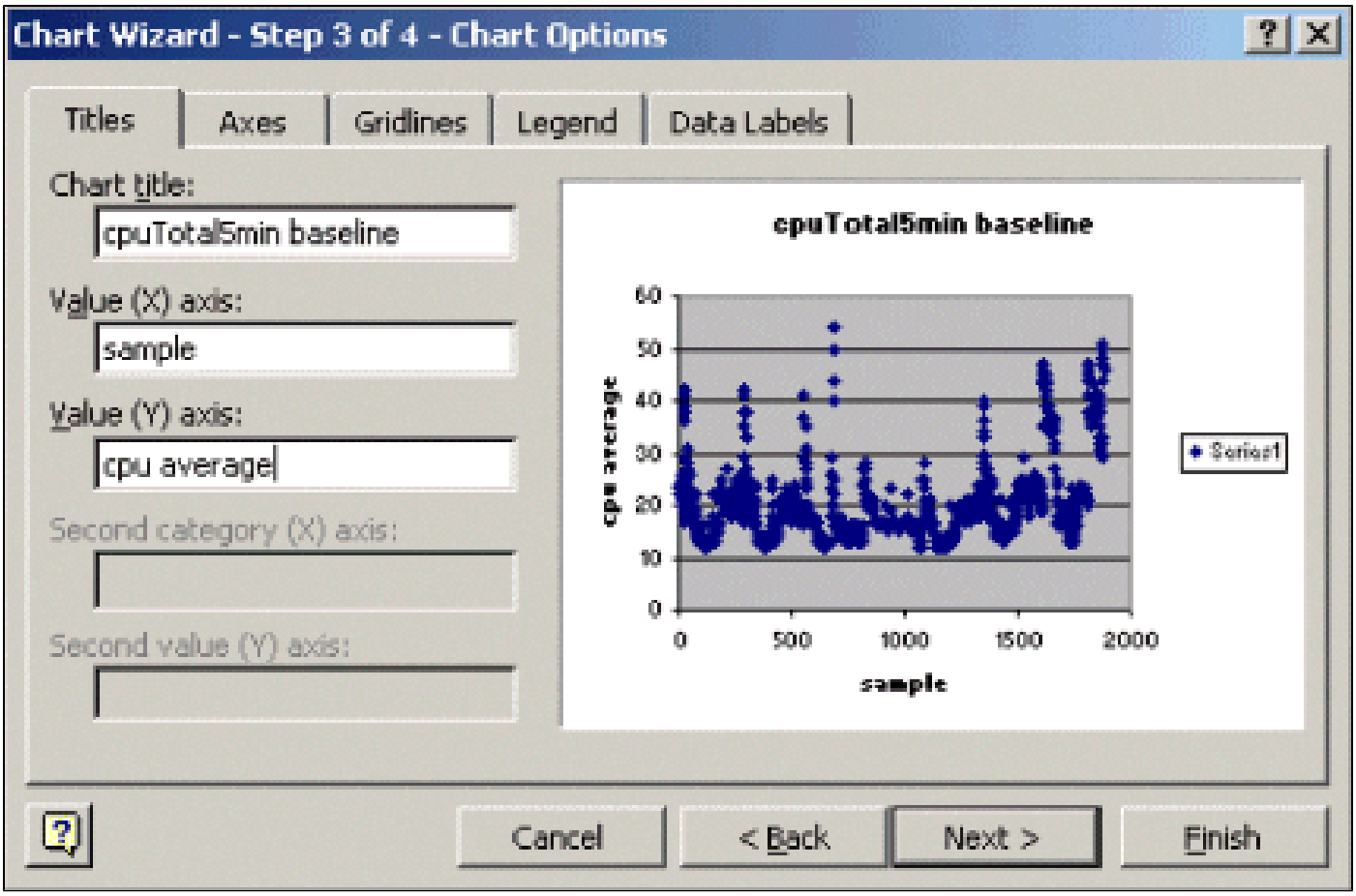


チャート ウィザード ステップ4では、新しいページor Existingページのオブジェクトとして散布図を適用するかを選択します。

目的の場所にグラフを配置する場合は、[Finish]をクリックします。

「What if」 分析

これで、散布図を使用して分析できます。ただし、続行する前に、次のことを実行する必要があります:

- ベンダーが推奨する MIB 変数のしきい値は何か（この例では、シスコがベンダー）。

  一般に、コア ルータは60の平均CPU使用率を超えないことを推奨します。シスコが 60 % を選択したのは、ルータにトラブルが発生したり、ネットワークに障害が起こった場合に、ルータに多少のオーバーヘッドが必要だからです。シスコはルーティング プロトコルが再計算されるか、または再コンバージェンスが、コア ルータは約40パーセントCPUオーバーヘッドが必要であると想定します。このパーセント値は、使用しているプロトコル、およびネットワークのトポロジや安定性によっても変わってきます。

- しきい値の設定に 60 % を使用したらどうなるか。

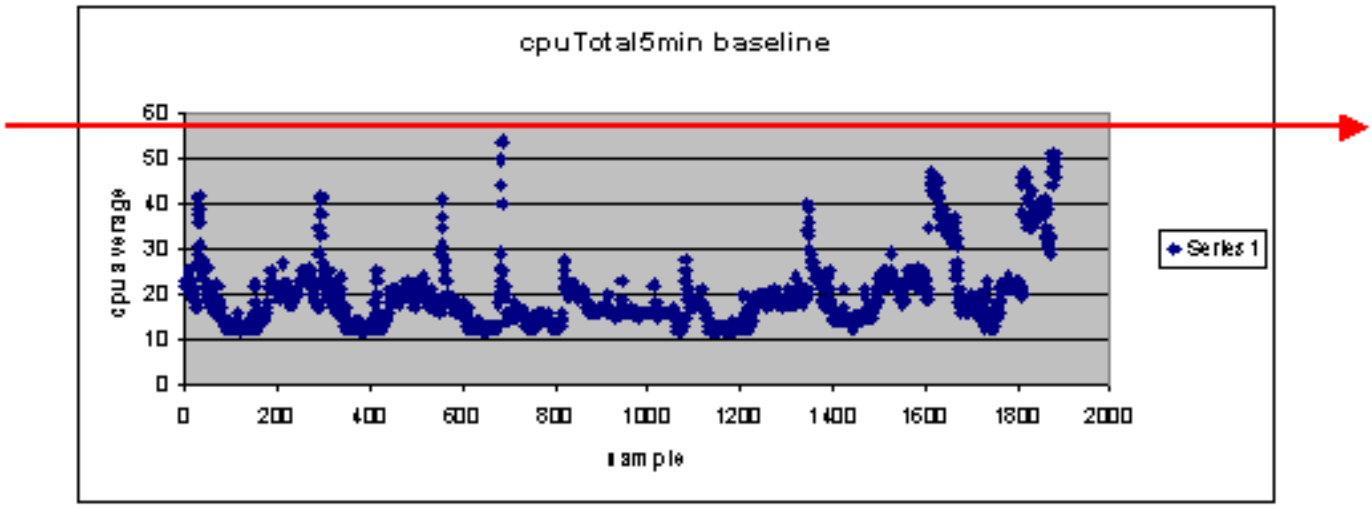
  60で散布図にわたる回線レベルを引出せば、データ ポイントでも、CPU使用率は60を超えることがわかります。そのため、NMS 端末でしきい値を 60 に設定すると、ポーリングの間、しきい値のアラートが発生しません。パーセンテージの60は、ルータで使用できます。ただし、データ ポイントの一部が60に近いため、散布図になります。ルータが60のしき

い値に達すると、CPUが60削減するそのポイントまで達するとするための計画を持つように近づいていることを事前に確認できるか確認するには素晴らしく。

- しきい値を 50 % に設定したらどうなるか。

  このルータが4回このポーリング サイクルで50の使用率となり、しきい値アラームを常に生成すると想定されます。異なるしきい値を設定したらどうなるかを確認するためにルータ グループを表示させる場合、このプロセスの重要度が増します。たとえば「に設定されている場合、全コア ネットワーク向けの50でしきい値または」数 しきい値を 1 つだけ選択することが難しいことがわかります。

」分析、CPUしきい値「



しきい値を容易に設定する方法として、レディ、セット、ゴーのしきい値方法があります。この方法論では、連続して 3 つのしきい値を使用します。

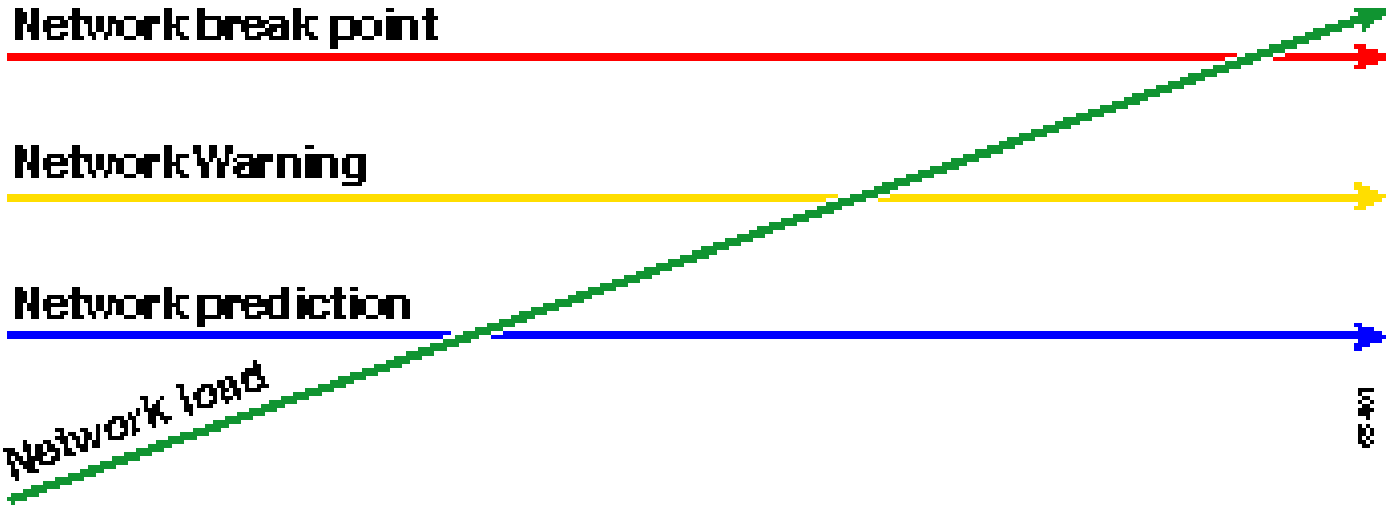
- レディ（Ready）：将来どのデバイスが注意を必要とする可能性が高いかを予測するために設定したしきい値
- セット（Set）：初期インジケータとして使用するしきい値。このしきい値によって、修理、再設定、またはアップグレードの計画を開始するように警告されます。
- ゴー：ユーザまたはベンダーが障害状態であると考えるしきい値。これを修復するには何らかのアクションが必要です。この例では60 %

次の表では、レディ、セット、ゴーの戦略について示しています。

しきい値	アクション	結果
45 %	さらに調査します	処置プランの[List of]オプション
50 %	アクションプランの策定	処置プランの手順のリスト
60 %	アクション プ	ルータではしきい値の超過がな

	ランを実装する	くなった。レディ モードに戻る

レディ、セット、ゴー方法論により、前述したオリジナルのベースライン チャートが変わります。次の図は、変更後のベースライン チャートを示しています。図の他の交点を識別する直前にした、計画して対応する多くの時間があります。



このプロセスでは、注意焦点を合わせられ、ネットワーク例外に他のデバイスにかかわられてに注意してください。デバイスがしきい値を下回っている場合、正常であることを前提としています。

次の手順を最初から考えているネットワークを長期間保持するようにプロビジョニングされます。このタイプの計画を実行することは、予算を計画する上でも有効です。最初の5が入るものルータ、中央の固定ルータを理解し、下部の対応ルータがある場合は、それらのルータに基づいてアップグレード何処置プラン オプションはどのくらい予算を計画で必要か簡単にできます。同じ戦略を Wide-Area Network（WAN）やその他の MIB OID にも使用できます。

ステップ5: 修正によって認識される差し迫った問題

ステップ 5 は、ベースライン プロセスの中でも簡単な部分の 1 つです。どのデバイスがゴーしきい値を超えたかを特定できれば、そのデバイスをしきい値以下に戻すアクション プランを作成します。

利用できるオプションについて、Cisco's Technical Assistance Center（TAC）までご相談いただくか、システム エンジニアにお問い合せください。しきい値を下げることに費用がかかるとは考えないでください。CPU の問題によっては、設定を変更してすべてのプロセスを効率的な方法で確実に実行すれば、解決できます。たとえば、アクセス コントロール リスト（ACL ）は、ルータのCPUにルータで実行できるパスが非常に高い原因でパケットを実行することができます。場合によっては、パケット スイッチング パスを変更し、CPU ACLの影響を軽減するにNetFlowスイッチングを実行できます。問題の内容を問わず、このステップでは全ルータのしきい値を設定値以下に戻すことが必要です。これにより、しきい値アラームが発生し過ぎて NMS 端末がフラッディングするリスクを負わずに、後でしきい値を設定できます。

ステップ6: しきい値モニタリングのテスト

このステップでは、プロダクション ネットワークで使用するツールにより、ラボでしきい値をテストします。しきい値をモニタリングするには、一般的に、2 つの方法があります。自分のネットワークに最適な方法を選択する必要があります。

- SNMP プラットフォームまたはその他の SNMP モニタリング ツールによって、ポーリングと比較を行う方法

  この方法では、トラフィックをポーリングするのにネットワークの帯域幅をより多く使用し、SNMP プラットフォームでのポーリング サイクルにも時間がかかります。

- ルータの Remote Monitoring (RMON; リモート モニタリング) アラームおよびイベントの設定によって、しきい値が超過した場合だけアラートが送られる方法

  この方法を使用すると、ネットワーク帯域幅の使用状況は減りますが、ルータでのメモリと CPU の利用率が増えます。

SNMPを使用してしきい値の実行

最初のポーリングを設定したときにHP OpenView NNM SNMPを使用する方式を設定するには、[Options > Data Collection & Thresholdsの順に選択してください。ただし今回は、収集のメニューで [Store, No Thresholds] を選択する代わりに、[Store, Check Thresholds] を選択します。しきい値を設定すると、複数のpingまたは複数のSNMPウォークを送信して、ルータのCPU使用率を発行できます。意図的に CPU の利用率がしきい値を超えるようにできない場合は、しきい値を下げる必要がある場合もあります。いずれの場合も、thresholdメカニズムが機能していることを確認する必要があります。

この方法の使用に関する制限の 1 つは、複数のしきい値を同時に設定できないことです。3 つの異なるしきい値を同時に設定するには、3 つの SNMP プラットフォームが必要になります。Concord Network HealthやTrinagy TRENDなどのツールを使用すると、同じOIDインスタンスに複数のしきい値を設定できます。

システムが一つのしきい値を同時に処理できる場合は、戦略にシリアル方式の対応のセットと見なすことができます。つまり、継続的にレディしきい値に達する場合、調査を開始して、そのデバイスのセット レベルまでしきい値を上げるということです。継続的にセット レベルに達する場合は、アクション プランを策定し、そのデバイスゴー レベルにまでしきい値を上げます。さらに、ゴーのしきい値に頻繁に達しているときは、アクション プランを実行します。この方法は、同時に 3 つのしきい値を設定する方法と同じように動作しますが、はSNMPプラットフォームのしきい値設定を変更できるように、少し時間がかかります。

RMONアラームおよびイベントを使用してしきい値の実行

RMON アラームとイベントの設定を使用すると、ルータ自体が複数のしきい値を監視できます。ルータはしきい値が超過している状態を検知すると、SNMP プラットフォームに SNMP トラップを送信します。トラップを転送するためには、各自のルータ設定に SNMP トラップ受信装置を設定する必要があります。アラームとイベントとの間には相互関係があります。アラームは、特定のしきい値について OID をチェックします。しきい値に達した場合、アラームのプロセスはいずれかのSNMPトラップ メッセージEvent of process or RMONログ エントリを作成するための両方で起動します。このコマンドの詳細については、RMONアラームおよびイベント設定コマンド

[を参照してください](#)。

次のルータ設定コマンドを実行すると、ルータは 300 秒ごとに cpmCPUTotal5min を監視します。は、CPUが40に戻る場合CPUが60を超える場合、点灯し、イベント2を起動イベント1が。いずれの場合も、SNMPトラップ メッセージがプライベート コミュニティ ストリングを持つNMSステーションに送信されます。

レディ、セット、ゴーの方法を使用するには、次の設定文をすべて使用します。

```
rmon event 1 trap private description "cpu hit60%" owner jharp
rmon event 2 trap private description "cpu recovered" owner jharp
rmon alarm 10 cpmCPUTotalTable.1.5.1 300 absolute rising 60 1 falling 40 2 owner jharp


rmon event 3 trap private description "cpu hit50%" owner jharp
rmon event 4 trap private description "cpu recovered" owner jharp
rmon alarm 20 cpmCPUTotalTable.1.5.1 300 absolute rising 50 3 falling 40 4 owner jharp

rmon event 5 trap private description "cpu hit 45%" owner jharp
rmon event 6 trap private description "cpu recovered" owner jharp
rmon alarm 30 cpmCPUTotalTable.1.5.1 300 absolute rising 45 5 falling 40 6 owner jharp
```

次の例では、上記ステートメントによって設定された show rmon alarm コマンドの出力を示します。

<#root>

zack#

sh rmon alarm

```
Alarm 10 is active, owned by jharp
 Monitors cpmCPUTotalTable.1.5.1 every 300 second(s)
 Taking absolute samples, last value was 0
 Rising threshold is 60, assigned to event
1
 Falling threshold is 40, assigned to event
2
 On startup enable rising or falling alarm
Alarm 20 is active, owned by jharp
 Monitors cpmCPUTotalTable.1.5.1 every 300 second(s)
 Taking absolute samples, last value was 0
 Rising threshold is 50, assigned to event
3
 Falling threshold is 40, assigned to event
4
 On startup enable rising or falling alarm
Alarm 30 is active, owned by jharp
 Monitors cpmCPUTotalTable.1.5.1 every 300 second(s)
 Taking absolute samples, last value was 0
 Rising threshold is 45, assigned to event
5
 Falling threshold is 40, assigned to event
6
```

```
 On startup enable rising or falling alarm
```

次の例では、show rmon event コマンドの出力を示します。

```
<#root>

zack#

sh rmon event

Event 1 is active, owned by jharp
 Description is cpu hit60%
 Event firing causes trap to community
private, last fired 00:00:00
Event 2 is active, owned by jharp
 Description is cpu recovered
 Event firing causes trap to community
private, last fired 02:40:29
Event 3 is active, owned by jharp
 Description is cpu hit50%
 Event firing causes trap to community
private, last fired 00:00:00
Event 4 is active, owned by jharp
 Description is cpu recovered
 Event firing causes trap to community
private, last fired 00:00:00
Event 5 is active, owned by jharp
 Description is cpu hit 45%
 Event firing causes trap to community
private, last fired 00:00:00
Event 6 is active, owned by jharp
 Description is cpu recovered
 Event firing causes trap to community
private, last fired 02:45:47
```

この両方の方法を試して、どちらの方法が自分の環境に最適であるかを確認します。両方の方法を組み合わせて順調に動作する場合もあります。いずれの場合も、すべてが正しく機能するため、テストは、ラボ環境で実行する必要があります。ラボでテストした後で、ルータの小さなグループに限定的な導入はオペレーション センターにアラートを送信するプロセスをテストすることができます。

この場合、プロセスをテストするためのしきい値を下げる必要があります。実稼働ルータで意図的にCPUを上げることは推奨されません。また、アラートがオペレーション センターの NMS 端末に着信すると、装置がしきい値を超過したときに確実に通知される報告ポリシーが存在することを確認する必要があります。この設定は、Cisco IOS バージョン 12.1(7)を使用して、ラボで試験を行いました。問題が発生する場合は、IOSバージョンで不具合がある場合は、シスコの設計、またはシステム エンジニアに確認する必要があります。

ステップ7: SNMP またはRMON を使用したしきい値モニタリングの実施

ラボ、または限定された実装の中で、しきい値モニタリングのテストが完全に終わったら、コア

ネットワーク全体にしきい値を設定する準備ができています。これで、バッファ、空いているメモリ、Cyclic Redundancy Check（CRC; サイクリック冗長性検査）エラー、ATM のセル損失などをはじめとする、ネットワーク内にある他の重要な MIB 変数について、このベースライン プロセスを体系的に実行できます。

RMONアラームおよびイベント設定を使用すると、NMSステーションから、ポーリングを停止できます。この結果、NMS サーバ上の負荷が減り、ネットワーク上のポーリング データ量も低減します。重要なネットワークのヘルス インジケータにこのプロセスで組織的に移動することによって、ネットワーク アプライアンスでRMONアラームおよびイベントを使用して自身を監視するポイントに簡単になる可能性があります。

追加 MIB

このプロセスを学習すると、他のMIBを基準に調査し、モニタすることができます。次のサブセクションで、いくつかの役に立つ OID の簡単なリストと説明を示します。

ルータの MIB

メモリの特性は、ルータの状態を決定する際に非常に役立ちます。十分なルータが通常動作する使用可能なバッファ領域が必要です。ルータがバッファ スペースが不足し始めると、新しいバッファを作成し、着信および発信パケットのバッファを検索するためにCPUが困難に動作です。バッファに関する詳細な解説は、このドキュメントの対象外です。ただし、通常、十分なルータはもしも、小数のバッファ ミス、バッファ障害または空きメモリがゼロの状態はありません。

Object	説明	OID
ciscoMemoryPoolFree	管理対象装置で現在、未使用のメモリプール	1.3.6.1.4.1.9.9.48.1.1.1.6

	のバイト数	
ciscoMemoryPoolLargestFree	現在、未使用のメモリプールの隣接最大バイト数	1.3.6.1.4.1.9.9.48.1.1.1.7
bufferElMiss	バッファ要素のミス数	1.3.6.1.4.1.9.2.1.12
bufferFail	バッファ割り当て失	1.3.6.1.4.1.9.2.1.46

	敗数	
bufferNoMem	空きメモリ不足によるバッファの作成障害数	1.3.6.1.4.1.9.2.1.47

Catalyst スイッチ MIB

Object	説明	OID
cpmCPUTotal5min	最後の 5 分間に CPU がビジー状態であった合計パーセント。このオブジェクトは、OLD-CISCO-SYSTEM-MIB の avgBusy5 オブジェクトを推奨しない。	1.3.6.1.4.1.9.9.109.1.1.1.5
cpmCPUTotal5sec	最後の 5	1.3.6.1.4.1.9.9.109.1.1.1.3

	秒間に CPU がビジー状態であった合計パーセント。このオブジェクトは、OLD-CISCO-SYSTEM-MIB の busyPer オブジェクトを廃棄する。	
sysTraffic	前回のポーリング期間の帯域利用率のパーセント	1.3.6.1.4.1.9.5.1.1.8
sysTrafficPeak	最後にポート カウンタがクリアーされてから、またはシステムが起動してからのトラフィック メータの最高値	1.3.6.1.4.1.9.5.1.1.19
sysTrafficPeaktime	メータ最高値の発生後の時間（100 分の 1 秒単位）	1.3.6.1.4.1.9.5.1.1.20
portTopNUtilization	システムのポート利用率	1.3.6.1.4.1.9.5.1.20.2.1.4
portTopNBufferOverFlow	システム	1.3.6.1.4.1.9.5.1.20.2.1.10

	のポートのバッファ オーバーフロー数	

シリアル リンク MIB

Object	説明	OID
locIfInputQueueDrops	入力キューがいっぱいのためドロップされたパケット数	1.3.6.1.4.1.9.2.2.1.1.26
locIfOutputQueueDrops	出力キューがいっぱいのためドロップされたパケット数	1.3.6.1.4.1.9.2.2.1.1.27
locIfInCRC	冗長性チェックサム エラーが発生した入力パケット数	1.3.6.1.4.1.9.2.2.1.1.12

RMON アラームおよびイベント設定コマンド

アラーム

RMON アラームは、次の構文によって設定できます。

<#root>

```
rmon alarm number variable interval {delta | absolute} rising-threshold value
           [event-number] falling-threshold value [event-number]
           [owner string]
```

要素	説明
番号	アラーム番号。RMON MIB のアラーム テーブル内にある alarmIndex と同一である。
可変	監視対象の MIB オブジェクト。RMON MIB のアラーム テーブルで使用される alarmVariable に変換する。
interval	アラームが MIB 変数を監視する時間（秒単位）。RMON MIB のアラーム テーブルで使用される alarmInterval と同一である。
delta	MIB 変数間の変更をテストする。RMON MIB のアラーム テーブルの alarmSampleType に影響を与える。
絶対	各 MIB 変数を直接、テストする。RMON MIB のアラーム テーブルの alarmSampleType に影響を与える。
rising-threshold value	アラームが起動する値
event-number	（オプション）上昇しきい値または下降しきい値が、制限値を超えたときに起動するイベント番号。この値は、RMON MIB のアラーム テーブルの alarmRisingEventIndex または the alarmFallingEventIndex と同一である。
falling-threshold value	アラームがリセットされる値
owner string	（オプション）アラームの所有者を示す。RMON MIB のアラーム テーブルの alarmOwner と同一である。

イベント

RMON イベントは、次の構文によって設定できます。

<#root>

```
rmon event number [log] [trap community] [description string]
           [owner string]
```

要素	説明
番号	割り当てられているイベント番号。これは、RMON MIB の eventTable 内にある eventIndex と同一である。
log	（オプション）イベントが起動し、RMON MIB の eventType を log または log-and-trap に設定するときに、RMON ログ エントリを作成する。
trap community	（オプション）このトラップに使用される SNMP コミュニティ ストリング。この行の RMON MIBのeventTypeの設定をSNMPトラップまたはログ、トラップ設定します。この値は、RMON MIB の eventTable 内の eventCommunityValue と同一である。
description string	（オプション）イベントの説明を示す。RMON MIB のアラーム テーブルのイベントの説明と同一である。
owner string	（オプション）このイベントの所有者を示す。RMON MIB のアラーム テーブルの eventOwner と同一である。

RMON アラームとイベントの実装

RMONアラームおよびイベントに関する詳細は、ネットワーク管理システムのベスト プラクティスのホワイト ペーパーのRMON Alarm and Event Implementationセクションを参照してください。

関連情報

- テクニカルサポートとドキュメント - Cisco Systems

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。