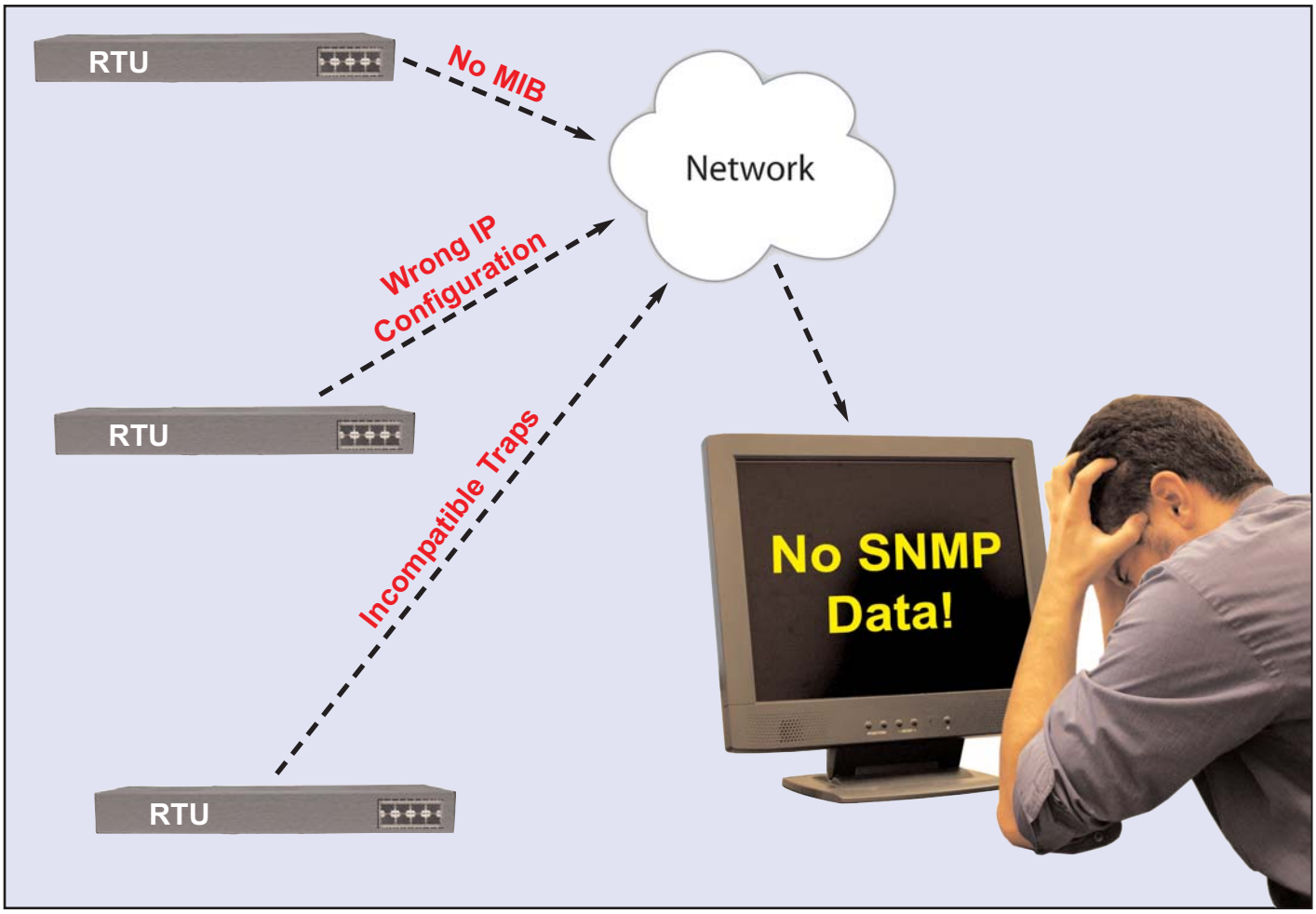
DPS Telecom

"Your Partners in Network Alarm Management"

Practical Guide to SNMP Troubleshooting



You don't need to be an SNMP expert to monitor your network effectively, but you will have to troubleshoot some common issues.

This white paper is a practical guide to SNMP troubleshooting, focusing on the problems that arise and how you can solve them.

Version 1.0
Released November 7, 2006

Printed in the U.S.A.

Executive Summary

Using SNMP to monitor your network can be a powerful tool for increasing your network reliability. Unfortunately, SNMP implementations are often hampered by a variety of different problems.

This SNMP troubleshooting guide will help you to identify and solve a variety of SNMP issues.

Inside, you will find powerful tips for identifying and solving problems with your SNMP devices, traps, network, manager, MIB files, and more. With this key information, you can eliminate hours of frustrating SNMP troubleshooting from your busy schedule.

Contents

SNMP Fundamentals

This section of the guide is a brief overview of several core SNMP concepts. **To start troubleshooting SNMP issues immediately, simply jump ahead to page 6.**

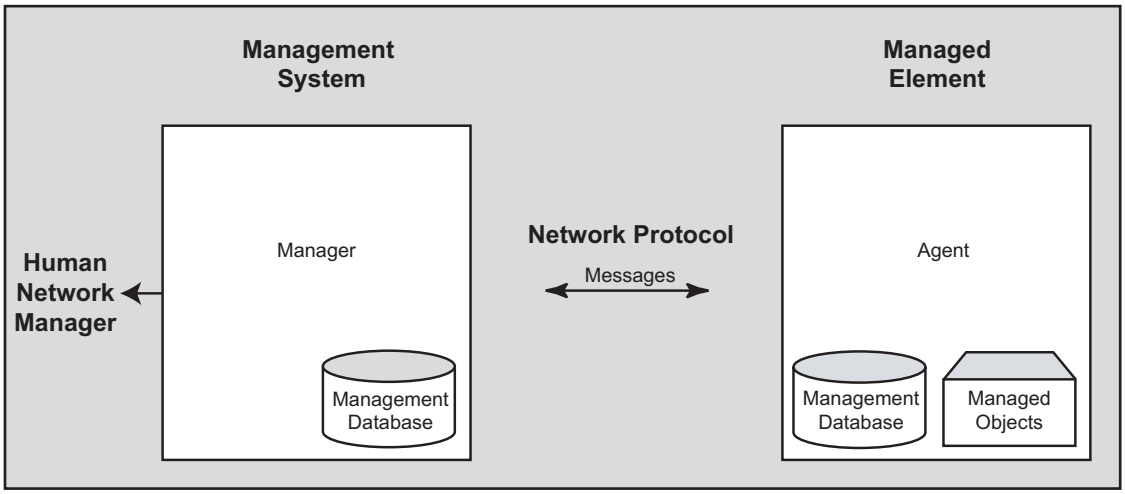
A Brief History of SNMP

Since its creation in 1988 as a short-term solution to manage elements in the growing Internet and other attached networks, SNMP has achieved widespread acceptance.

SNMP was derived from its predecessor, SGMP (Simple Gateway Management Protocol), and was intended to be replaced by a solution based on the CMIS/CMIP (Common Management Information Service/Protocol) architecture. This long-term solution, however, never received the widespread acceptance of SNMP.

SNMP Architecture

SNMP is based on the manager/agent model, consisting of a manager, an agent, a database of management information, managed objects, and the network protocol. The manager provides the interface between the human network manager and the management system. The agent provides the interface between the manager and the physical device(s) being managed.



SNMP uses a manager/agent architecture. Alarm messages (Traps) are sent by the agent to the manager

The manager and agent use a Management Information Base (MIB) and a relatively small set of commands to exchange information. The MIB is organized in a tree structure with individual variables, such as point status or description, being represented as leaves on the branches. A long numeric tag or object identifier (OID) is used to distinguish each variable uniquely in the MIB and in SNMP messages.

Understanding the OID

OIDS are unique "object identifiers." They are sequences of numbers separated by decimal points (e.g., 1.3.6.1.4.1.2682.1). The MIB associates each OID with a readable label (e.g., dpsRTUAState) and various other parameters related to the object. The MIB serves as a data dictionary or codebook that is used to assemble and interpret SNMP messages.
For example, when an SNMP manager wants to know the value of an object/characteristic, such as the state of an alarm point, the system name, or the element uptime, it will assemble a GET packet that includes the OID for each object/characteristic of interest. The element receives the request and looks up each OID in its code book (MIB). If the OID is found (the object is managed by the element), a response packet is assembled and sent with the current value of the object / characteristic included. If the OID is not found, a special error response is sent that identifies the unmanaged object.

Alarm Messages Over SNMP

SNMP typically uses five basic messages (Get, GetNext, GetResponse, Set and Trap) to communicate between manager and agents.

The Get and GetNext messages allow the manager to request information for a specific variable. The agent, upon receiving a Get or GetNext message, will issue a GetResponse message to the manager with either the information requested or an error indication as to why the request cannot be processed.

A Set message allows the manager to request a change be made to the value of a specific variable, such as in the case of an alarm remote that will operate a relay. The agent will then respond with a GetResponse message indicating the change has been made or an error indication as to why the change cannot be made.

The Trap message allows the agent to spontaneously inform the manager of an "important" event.

The Inform, sometimes supported in v2c or v3 implementations, is like a trap that also requires a confirmation response from the SNMP manager. This trades network resources for reliability and is discussed in more detail on page 7.

How SNMP v2c and Reliable Inform Notifications Can Help You

SNMP v2c offers a variety of improvements over v1. One significant advantage is the ability to use the new Inform notification in place of traps.

Trap notifications, supported in SNMP v1 and later, are considered non-robust because the SNMP manager doesn't send an acknowledgement in response to the Trap. The device sending the Trap sends it only once. The sending device has no confirmation that the Trap has been received, so there's no guarantee that the alarm information has been successfully sent to the SNMP manger.

Inform notifications, supported in some SNMP v2c implementations or SNMP v3, are designed for confirmed delivery. When an SNMP manager receives an Inform, it sends a confirmation response back to the SNMP agent device. If the SNMP agent device doesn't receive a response to its Inform, it resends it until the SNMP manager sends the confirmation response or the specified number of retries is exhausted.

But Inform notifications are not the best choice for everybody; you're choosing a trade-off of reliable delivery at the cost of heavier network traffic. A Trap is only sent once, which places only a very small demand on network resources. The Inform requires a response packet to be sent (and messages may be sent several times if responses are not received), which increases reliability at the cost of increased network traffic.

Looking for More Information?

Additional information about SNMP fundamentals is beyond the scope of this white paper. If you're new to SNMP and would like a more detailed introduction, please read *How to Implement SNMP Monitoring in Your Network: A Practical, Step-By-Step Guide*, another white paper available at no cost from DPS Telecom (www.dpstele.com/white-papers/snmp-implementation).

3 SNMP Trap Issues That Can Disrupt Your Monitoring

Some SNMP problems are caused by the content of the SNMP traps being sent. Because identifying these issues is a fairly quick process, it is a good idea to look for them before moving on to more time-intensive procedures. Be sure to check for these trap issues as you begin troubleshooting.

1) Incompatible Trap Versions

If your SNMP manager is configured to accept v1 traps and your device is sending v2 traps, you will encounter problems. Similarly, some managers that are configured to receive v2 traps will not correctly parse v1 traps. Configure your RTU to send the version of traps that your manager is setup to accept, or configure your manager to receive the type of traps that your remote equipment is sending. Generally speaking, most v2 managers can be configured to receive v1 traps.

2) Non-Standard Trap Formats

SNMP managers can also run into trouble if a device is sending non-standard traps. Although SNMP is a standard protocol, some people have modified the formats of their traps to suit special needs. They might have, for example, added an extra field to their traps to transmit a particular piece of additional data. If this change was not properly documented, it can cause trouble later.

Because this is not a very common SNMP issue, it tends to be one of the more difficult to identify. If you find yourself with a stubborn SNMP problem, don't forget to check for non-standard trap formats/content.

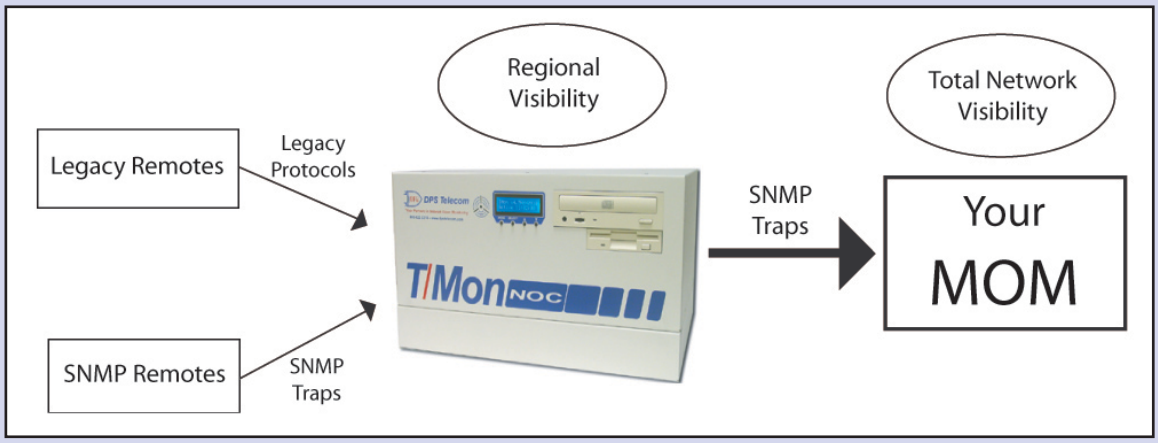
3) Altered Community Names

In most SNMP implementations, the community name used by the devices and the manager is "public." Some IT departments, however, have set up unique community names on their networks. This can cause trouble with your SNMP traps because some SNMP managers will use the community name as a unique identifier. If your manager is expecting "public" but finds a customized community name instead (or vice versa), it may simply discard the trap.

Another potential problem is switches that utilize variable community names. Devices connected to Shelf 1 might be given the community name "public-1", those on Shelf 2 given "public-2", etc. Unless you have a proprietary master that is expecting traps with variable community names, it may not handle them properly.

Check for any altered community names and make any necessary adjustments. Remember that **community names must match exactly and are case-sensitive.**

Bridge the Gap Between Protocols with T/Mon NOC



T/Mon NOC integrates legacy protocols into modern SNMP monitoring systems. You don't have to scrap your older equipment, delay SNMP migration, or suffer with multiple incompatible monitoring systems. T/Mon supports a wide range of legacy protocols in addition to SNMP traps, providing complete visibility on a single display. For larger network configurations, T/Mon mediates legacy protocols and can forward alarms as traps to your SNMP-compatible MOM.

5 Common MIB Issues

Even if SNMP traps are being properly sent from an agent, a SNMP manager with any of the following MIB issues will create errors and reduce your network visibility, increasing the chance of a costly outage.

1) Needed MIB Not Compiled Onto Your SNMP Manager

A MIB file is a sort of "codebook" that is required to interpret traps sent from your SNMP devices. Without the appropriate MIB, your SNMP manager will not be able to handle incoming traps from an SNMP device. Remember that:

- MIB files are generally available from your device manufacturer
- You must compile any new MIBs into your SNMP manager
- In some cases, device manufacturers will not provide MIBs for their devices. This is typically an attempt to force purchases of their proprietary SNMP manager or other equipment. One good way to work around this issue is to use a device that accepts manual input of trap values, such as the T/Mon NOC.

Essential SNMP: What is the MIB?

The MIB lists the unique object identifier (OID) of each managed element in an SNMP network. Your SNMP manager can't monitor your devices unless it has compiled their MIB files. The MIB is also a guide to the capabilities of your SNMP devices. For example, if your device's MIB lists OIDs for Traps but not for GetResponse messages, you know it will report alarms, but will not respond to alarm polls. Before compiling, MIBs can be modified in any text editor, such as Notepad or Wordpad. Learning to read MIBs can be difficult, but it's worth the trouble.

2) Incompatible MIBs

The two most common MIB types are DOS MIBs and UNIX MIBs. DOS MIBs may not work with a UNIX SNMP manager, and vice versa. Check to be sure that you are using MIBs that are compatible with your manager.

3) Missing Reference MIBs

Most main MIBs require additional reference (RFC) MIBs during compiling. If any of these RFC MIBs are missing, the main MIB will not compile properly. When compiling a MIB on DPS Telecom's T/Mon NOC, an error message is added to the MIB Manager log that indicates which MIB files are missing. You may access this log from the Report Viewer. Error reporting on other SNMP managers will vary, but you can always get a list of required reference MIBs by reading the main MIB.

```
DPS-MIB-V38 DEFINITIONS ::= BEGIN
IMPORTS
 DisplayString
      FROM RFC1213-MIB
 OBJECT-TYPE
      FROM RFC-1212
 enterprises
      FROM RFC1155-SMI;
```

The above example is taken from the beginning of a typical MIB file. The **bold** text identifies the three MIBs that are referenced in this main MIB. Remember that RFC MIBs will always be listed under "IMPORTS." Make sure that you have all RFC MIBs referenced by your main MIB and compile it again.

4) Typos in the MIB

Bad syntax in a MIB file can create errors when compiling. Exactly how much goes wrong will vary based upon the compiler that your are using. Some are less forgiving than others. Although typos in the MIB can take many forms, one of the most common is the incorrectly escaped file comment.

File comments in MIBs are offset from the rest of the file by double hyphens (--) and generally continue until the end of the line. An important exception is that comments can be ended by a second pair of hyphens on the same line. Any text on the same line after this second pair of hyphens will be parsed by the compiler as if it were normal MIB code, causing an error.

```
IMPORTS--This comment is correctly inserted
 DisplayString--As is this one--BUT THIS ONE WILL CAUSE AN ERROR!
      FROM RFC1213-MIB
```

As you can see in the above example, the first comment does not create a problem, nor does the second. The third, however, appears outside of the second double hyphen (--) on the same line and is considered part of the MIB code during compiling. The compiler will not know how to handle it, and an error will be generated. Look for this and other typos in your MIB files.

Get More From SNMP with Sets and Gets

Issuing Set and Get messages from your SNMP manager can enhance your monitoring and give you more control over your remote sites.

A Set message allows the manager to request a change be made to the value of a specific variable. The agent will then respond with a SetResponse message indicating the change has been made or an error indication as to why the change cannot be made. To be set, a variable must have read-writable characteristics defined in the MIB. Errors commonly occur when attempting to set a variable that has read-only attributes.

The Get and GetNext messages allow the SNMP manager to request information for a specific variable. The agent, upon receiving a Get or GetNext message, will issue a GetResponse message to the manager with either the information requested or an error indication as to why the request cannot be processed.

If you'd like additional information about MIB files to identify other kinds of typos, please read *Demystifying the MIB*, another white paper available at no cost from DPS Telecom (www.dpstele.com/white-papers/snmp-mib).

5) "Pre-compiled" MIBs

Using "pre-compiled" MIBs is not always the best choice. MIBs that were compiled for a target platform other than your manager can create a range of potential problems. The MIB files you use should be text-readable before you compile them to your manager.

Don't Forget the Obvious

You don't want to spend any unnecessary time searching for complex problems, so be sure to check for these simple and "obvious" ones during troubleshooting:

1) Is your remote device sending traps to the IP address of the target manager?

2) Is your remote device provisioned to send traps?

3) Is your IP addressing, including subnet and gateway, set up correctly on the remote device?

4) Is your IP routing configuration (DNS or static) correct?

6 Quick Steps to Identify and Solve Firewall Problems

Some SNMP problems are not directly caused by either manager or agent. The network connectivity between the two devices can sometimes be impeded by a firewall. **Firewalls that block UDP, SNMP, pings, or ports 161 or 162 are the most common issues.** Use the following steps to identify and solve firewall problems:

1) Ping a PC near the device

A simple ICMP ping to a PC near the device is a good initial test to determine connectivity status. If your pings to the PC are not returned, try pinging the gateway. Continue working your way up the network with your pings to identify the point where they stop. Check for firewalls, especially those that block UDP, SNMP, pings, or ports 161 or 162. Keep in mind that some networks block all ping traffic as a security measure.

Price is Only the First Part of Cost Justification — Make Sure Your Vendor Offers Guaranteed Results

Bob Berry
Chief Executive Officer
DPS Telecom

In my experience, clients who think hard about cost justification have a more important concern than just price. They want to make sure that they're not spending their company's money on a system that doesn't work as advertised.

That's smart. You have to be careful when working with equipment vendors, especially on protocol mediation projects. Most vendors can't support all your legacy equipment, and they don't have the development capabilities to make integration work.

Some vendors will charge you large NRE (non-refundable engineering) fees up front for custom work, and give no guarantee that the resulting product will meet your performance requirements.

Personally, I think that's a lousy way to do business. I give all my clients a 30-day guarantee: **If my product doesn't completely satisfy you, return it for a full refund.** If I can't give you a solution, I don't want your money. If I'm doing custom work for you, I don't expect you to pay for it until I've proven that it works to your satisfaction.

Very few vendors will make that guarantee. But you need to demand the best level of service from your vendor to ensure that your SNMP implementation is 100% successful.

2) Ping the device

Next, send another simple ICMP ping to the device to determine connectivity. If pings to the PC in Step 1 were successful, but pings sent to the device fail, the problem is almost certainly with your SNMP device.

3) Telnet and/or browse to the device

If the SNMP device you are testing supports Telnet connections or Web access, you should attempt to connect using one of these methods. If pings succeed but Telnet and/or browsing is blocked, this is a very good indication that you have a firewall issue.

4) Confirm the port configuration of the device

For additional security, some devices may use non-standard ports. If so, make sure that these ports are not blocked by a firewall and are accepted by the manager. Another potential solution is to reconfigure the device to use standard ports.

5) Confirm that important IP addresses are not blocked

A firewall may simply be blocking the IP address of your device and/or manager. Confirm that these or any other needed IP addresses are not being blocked.

6) Trace the route to the device

Tracing the "hops" that network traffic is following to reach the device can allow you to pinpoint a tricky firewall issue. A simple trace can be performed from the Command Prompt of Windows XP:

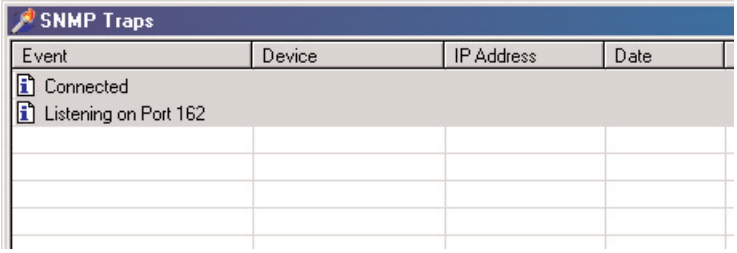
- •Open a Command Prompt in Windows XP
- •Type "tracert", a single space, and the IP address of the device you are trying to reach (i.e. "tracert 192.168.230.143")
- •Press return to start the trace
- •Show the output to your IT department to identify potential firewall problems

In-Depth Techniques to Identify and Solve Stubborn SNMP Problems

When the previous steps in this guide do not lead you to a solution, the following in-depth techniques can help you identify and solve tricky SNMP problems.

Use a network diagnostic utility to examine the paths of your SNMP traps

If your SNMP traps are not traveling over your network from SNMP agent to master, you need to pinpoint the location of the problem in order to solve it. The following utilities can help you to do this:



TrapManager (for v1 traps from DPS remotes)

DPS Telecom produces TrapManager, an efficient utility for analyzing SNMP v1 traps from DPS remotes on your network. Unlike less focused tools, this small software tool is one of the quickest methods to diagnose SNMP transport problems.

You may download TrapManager at no cost from DPS (www.dpstele.com/library/trapmanager.html).

To identify a transport issue on your network, you should run the following test on PCs in two specific network locations:

-First, as close to your manager as possible
-Second, as close to your DPS RTU as possible

Test Procedure:

•Install TrapManager on an appropriately located PC
•Configure your DPS SNMP remote to send v1 traps to Port162 of your PC running TrapManager
•Create an event that will send a trap
•Watch TrapManager on your PC to see if traps appear

If traps appear on the PC located near your SNMP manager, then traps are being sent from your RTU and traveling through your network to the location of your manager. In this case, the problem is almost always with your manager configuration. Check the settings of your SNMP manager.

If traps appear <u>only</u> on the PC located near your DPS RTU, then traps are being sent from your RTU and are being

obstructed on the way through your network. Check for firewall issues and other network problems.

If traps do not appear on either PC, then your SNMP remote is not properly sending traps. If your remote supports debug mode, you may use it to confirm this issue (details on page 13). Check the configuration of your SNMP RTU and adjust accordingly.

When you have finished this testing procedure, remember to return your device to its original configuration: reporting to the IP address of the primary manager.

Packet Sniffer and MIB Browser

No matter the manufacturer of your SNMP remote, a combination of a packet sniffer, such as Ethereal, and MIB Browser can help you troubleshoot a wide range of network problems.

Ethereal is a network protocol analyzer and is available at no cost from www.ethereal.com.

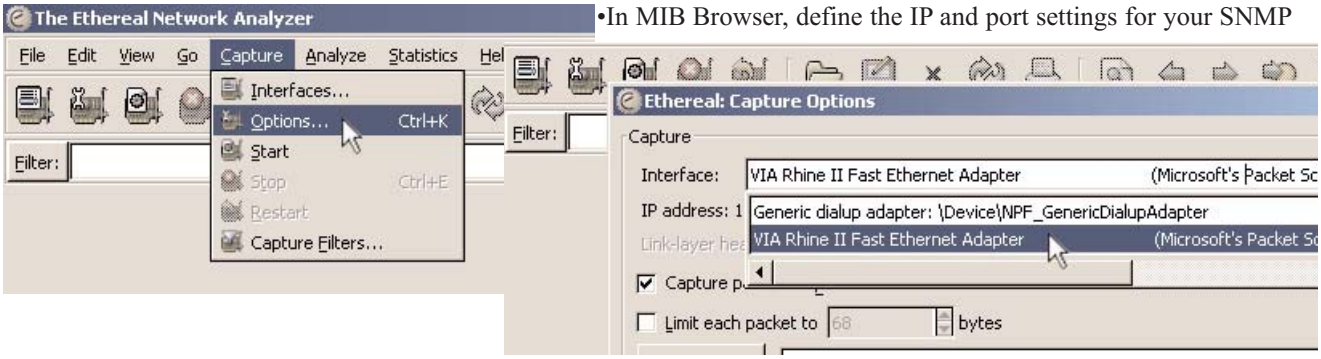
MIB Browser is an SNMP network management tool. A fully-functional 30-day trial is available at www.ireasoning.com.

•First, select a PC on your network that is as close to your SNMP master as possible.
Ideally, the PC and the SNMP master will be directly connected to the same hub. Hubs are better suited for this analysis than switches because they always send all received packets out on all ports. If you use a switch, your PC running Ethereal may not receive the packets that you need to perform your analysis.

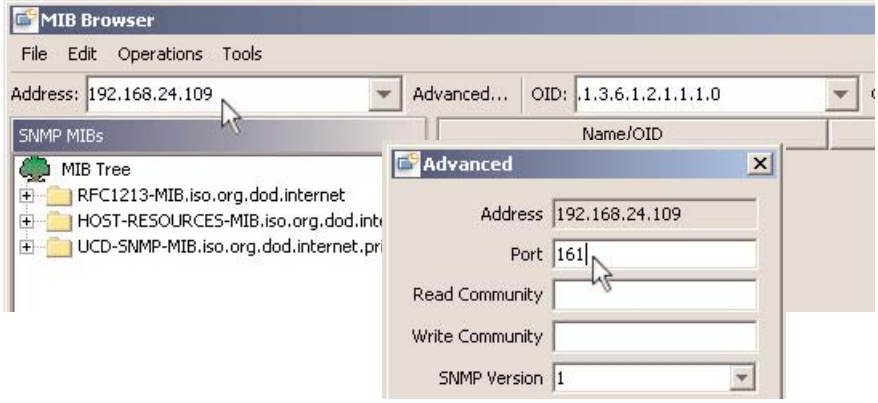
•Install Ethereal and MIB Browser onto the selected PC.

•Start Ethereal and MIB Browser.

•In Ethereal, assign your PC's network card (NIC) as the capture interface in the Capture Options window:

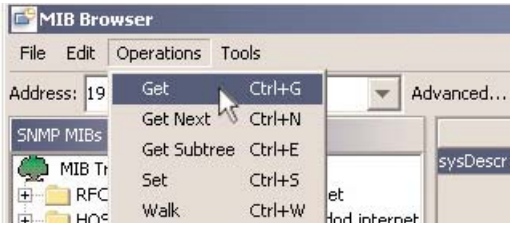


•In MIB Browser, define the IP and port settings for your SNMP remote.
-Input the IP address of your SNMP remote into the "Address:" field.
-Click the "Advanced..." button and specify the appropriate IP port (default is 161 for most SNMP devices) and any other necessary settings for your SNMP remote.



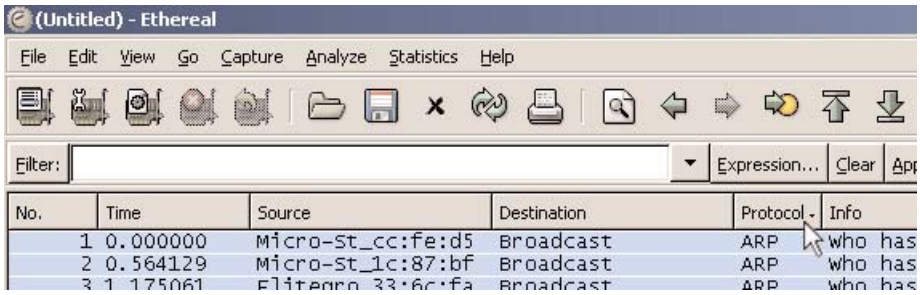
•Return to Ethereal and start a new Live Capture.

•In MIB browser, send a Get to your SNMP remote.
-Any GetResponses received will appear in the list on the right of the screen.
-If you receive no GetResponses, make sure that you have correctly specified the IP address and port number for your SNMP RTU. If you still do no receive GetResponses, you have a network problem or your RTU is not sending traps properly.



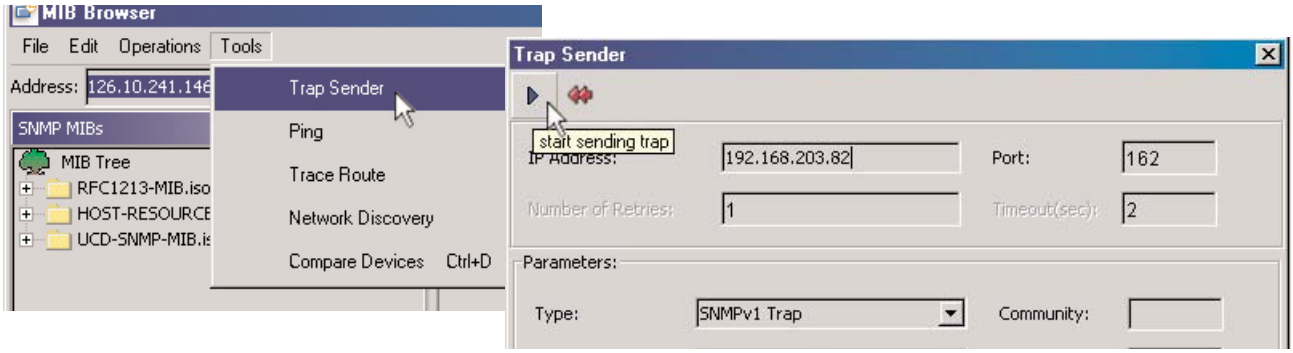
•Return to Ethereal and stop the Live Capture.

•Examine the packets captured by Ethereal, looking for SNMP in the "Protocol" column
-You can sort packets by protocol by click the "Protocol" column header to group all SNMP packets together



•**If your traps are incorrectly addressed,** they will never reach your manager.
-If traps are coming through but not displayed on your SNMP manager, your SNMP RTU may simply be sending traps to the wrong IP address. Check for this problem in the "Destination" IP address column in Ethereal.

•**If your traps appear to be correctly addressed,** you may have a problem with your SNMP manager configuration.
-To further isolate the problem to your SNMP manager, you may use the Trap Sender in MIB Browser to send test traps to your SNMP manager while you monitor its display. The Trap Sender is located in the Tools menu. Simply specify the IP address of your manager and a few other settings, then click the blue triangle button to send a test trap. If these test traps are not displayed by your SNMP manager, the problem is almost certainly with the configuration of your manager.



•**If you received no SNMP traps,** you must check your network and RTU configurations separately to determine the cause of the problem. One good next step is to identify another PC on the network that is as close to your SNMP remote as possible (ideally connected via Hub to ensure complete visibility in Ethereal). Install Ethereal and MIB Browser on that PC and repeat this procedure. By conducting the test as close to your RTU as possible, you will minimize the effect of firewalls and other potential network problems. If you still do not see traps in Ethereal after testing again, your SNMP remote may not be sending traps properly, or at all.

How to Verify That Your SNMP Device is Sending Traps

If traps aren't being properly sent from your SNMP device, they obviously won't make it to their destinations. Your SNMP device may have a debug mode option. Once in debug mode, you will see a display of traps being sent out by the device. By creating events at the site and watching the debug-mode trap display, you can determine whether traps are trying to be sent. Remember that you may need to reconfigure the remote to send a trap in response to the event you're generating. DPS Telecom remotes can be configured to send traps on a point-by-point basis.

Another testing method is to look for a coldstart trap following a bootup. Power cycle your device and watch for the trap that, on many SNMP devices, is sent when booting. Check your device documentation to ensure that you don't waste time looking for a coldstart trap that your device isn't designed to send.

If traps are not being sent, you have an issue with your SNMP device. If they are being sent, check your network setup, firewalls, MIB files, and trap formats using the other sections of this guide.

Easy SNMP debugging with the NetGuardian

If you have a NetGuardian RTU, you can easily verify that it is sending traps correctly:

1) Open a Telnet connection to the NetGuardian in HyperTerminal or other Telnet-capable software

2) Enter your NetGuardian password to log in

3) Press 'd' to reach the debug menu

4) Press 'r' to enter Report Mode

5) Generate an event at the site that would normally trigger an SNMP trap

6) Watch for SNMP traps to be reported in your Telnet session. If traps are properly shown in Report Mode, the problem is not with your NetGuardian RTU.

DPS Telecom Tech Support for Your SNMP Problem

The NetGuardian, the T/Mon NOC, and every other DPS Telecom product includes comprehensive technicial support. If you've purchased a DPS product and are encountering SNMP issues, contact DPS Tech Support today at 559-454-1600.

At DPS Telecom, the representative who answers your call isn't an intern reading from a script. **DPS Tech Support representatives are engineers** who contribute to product development. And, if your problem requires additional expertise, the DPS Engineering Department that designed your product is right down the hall.

Chris Hower
Tech Support at DPS

Success Story: UBTA-UBET Makes One-Step Upgrade to SNMP

T/Mon Makes Legacy RBOC Equipment Work for a Small Modern Telco

UBTA-UBET plant manager Rick Hofmann.

When Utah independent telco UBTA-UBET bought three Qwest exchanges in 2001, the deal included some extras — Qwest's legacy E2A alarm shelves and former Qwest telemetry tech Rick Hofmann.

After 22 years with Qwest, Hofmann's new job presented him with a tough challenge. How was he going to adapt his RBOC alarm equipment to the operations of a much smaller company? Qwest routed all alarms to a central NOC in Colorado; the lightly-staffed UBTA-UBET needed a local system that would automate alarm monitoring as much as possible.

Hofmann needed more than a technical solution — he needed a whole new way to monitor alarms.

RBOC centralized monitoring didn't work well for a small independent

As part of the acquisition deal, Qwest continued to monitor the three exchanges for another year, and Hofmann saw first-hand that RBOC-style centralized management didn't meet UBTA-UBET's needs.

"Qwest monitoring wasn't really workable for us," said Hofmann, who is now plant manager for UBTA-UBET. "Alarms went to the Qwest NOC in Colorado, and they would generate a trouble ticket, decide whether the alarm needed action and then call us. They wanted one point of contact, so we'd have them call our dispatch center, and they would call the tech."

Hofmann had five objectives for a new monitoring system

Hofmann's goal was an alarm system that would both support UBTA-UBET's existing equipment and provide tools for the company to effectively monitor alarms on its own.

Getting support for the E2A shelves was important, but simplifying and automating alarm handling was even more important. "Pager notification was my first objective. Support for legacy equipment was second," said Hofmann.

Hofmann listed his five objectives for the new alarm monitoring system:

1. Automatic alarm notification by pager and email.
2. Legacy support for the E2A shelves.
3. Access for dispatch center personnel to view alarms
4. Detailed notification.
5. Support for ASCII alarms and SNMP trap processing.

Solution: T/Mon bridges between legacy support and future capabilities

After surveying the market and looking at how telecom companies had modernized their alarm monitoring, Hofmann decided that an IAM-5 running the T/Mon Remote Alarm Monitoring System best met his objectives.

"T/Mon met all my requirements, while other systems didn't actually meet them all," said Hofmann. "Other systems might be able to do a page, but the notification wouldn't give you any alarm detail. Or they wouldn't expand to do some of the other functions we needed, like SNMP."

Results: More effective monitoring of more network elements

Since the T/Mon was installed, it has met all of Hofmann's objectives:

Pager Notification

"The main source of our alarm notification is via email or paging," said central office tech Richard Bell, who now is the primary manager of the T/Mon system. "We don't need to have someone physically watch the T/Mon screen all day, and after-hours and weekends, we depend completely on paging."

Legacy E2A Support

UBTA-UBET ordered its T/Mon unit with the optional E2A Interrogator software module and a 202-to-RS232 converter shelf. With these additions, "T/Mon supported the E2A shelves right out of the box," Hofmann said.

Dispatch Center Alarm Access

"The dispatch center can get into the T/Mon system and check alarms every 30 minutes, and make sure that the technicians are working on the problems," said Hofmann.

Detailed Notification

"The notification of alarms and the detail it gives you is the best thing T/Mon can do for you," said Bell. "Before we had the T/Mon, if a tower light went out, Qwest would call us and all they could say was 'You have a tower light out.' They didn't know where the tower was — there was no way of knowing. Now, when T/Mon pages you, it tells you it's this tower light, at this location, with this longitude and latitude, and then it gives you the FAA's phone number. And with our service area, which is 180 miles in diameter, you want to know what's wrong without wasting time or going on site."

ASCII and SNMP Support

T/Mon's ASCII and SNMP capabilities are helping UBTA-UBET move to a more modern system and monitor more network elements. UBTA-UBET now monitors its switches using T/Mon's ASCII alarm processing, which provides much more detailed alarm notification than major-minor discrete alarms. UBTA-UBET is also gradually implementing SNMP trap processing for new equipment.

Future Plans

After two years UBTA-UBET has used T/Mon, the company's alarm handling has substantially improved, said Hofmann. "The T/Mon has given us better notification, and I think it's made us a lot more responsive to alarms," Hofmann said.

Hofmann and Bell said they plan to use T/Mon to extend UBTA-UBET's alarm monitoring capability further in the future, expanding T/Mon coverage to all of the company's network and implementing analog monitoring of environmentals.

Reality Check: 8 Features That SNMP Managers Can't Match

1. **Detailed alarm notifications in plain English that your staff will immediately understand and take action on.** Every notification includes full information about the alarm, including its severity, location, date/time stamp, and a user-defined description.
2. **Immediate notification of changes of state (COSs)**, including new alarms and alarms that have cleared. You don't have to hunt to find out what's changed in your network — T/Mon lists it for you.
3. **A continuously updated list of all current standing alarms.** Even if the system operator acknowledges the alarm, it remains in the Standing Alarms screen until it is cleared.
4. **Text message windows displaying specific instructions for the appropriate action for an alarm.** System operators, even without extra training, will know precisely what to do and who to call in case of an alarm.
5. **Nuisance alarm filtering.** Unimportant alarms that generate meaningless status notices or oscillate between alarm and clear conditions subconsciously train your staff to ignore the alarm monitoring system. T/Mon filters out nuisance alarms, allowing your staff to focus its attention on serious threats.
6. **Pager and e-mail notifications.** Send alarm notifications directly to maintenance personnel, even if they're away from the NOC.
7. **Derived alarms and controls** that combine and correlate data from multiple alarm inputs and automatically control remote site equipment to correct complex threats.
8. **Mediation of all alarms to SNMP traps sent to MOM**

The T/Mon NOC Remote Alarm Monitoring System provides total visibility of your network status and automatically notifies the right people to keep your network running.

Sign up for a Web demo of T/Mon NOC at www.dpstelecom.com/webdemo

SNMP Glossary of Key Terms

Agent: A hardware device or software program that reports to an SNMP manager. In network alarm management, an SNMP agent is typically an RTU, but other network devices like switches, routers and hubs can also act as SNMP agents. An SNMP agent can also be a subsection of a larger device, like the SNMP Agent software module in T/MonXM, which mediates T/Mon alarms to SNMP traps.

Community string: An SNMP security password. There are three kinds of community strings:

Read Community: Allows an SNMP manager to issue Get and GetNext messages.

Write Community: Allows an SNMP manager to issue Set messages

Trap Community: Allows an SNMP agent to issue Trap messages.

Compiling: The process of importing a MIB file into an SNMP manager. To compile properly, a MIB file must be formatted in a text file according to the Structure of Management Information (SMI) standard.

COS (Change of State) alarm: A telemetry alarm that is clearly labeled as reporting a change in status from clear to alarm or from alarm to clear.

Event: In SNMP terms, any change of status in a managed object in the network. SNMP equipment can generate traps for many different kinds of events, not all of which are important for telemetry. The ability to filter unimportant events is essential for high-quality SNMP alarm management

Get: An SNMP message issued by a manager that requests the status of a managed object.

GetNext: An SNMP message issued by a manager, used to walk down a range of OIDs. The GetNext request retrieves the value of the managed object one number after the OID listed in the request.

GetResponse: SNMP message issued by an agent in response to a Get, GetNext or Set request from the SNMP manager.

Inform Notification: An SNMP message (supported in some v2c and v3 implementations) that is similar to a trap but requires a confirmation response from the manager. This is more robust than a standard trap and offers better reliability, but it also consumes more network resources.

Internet Protocol (IP): the network layer datagram protocol of the TCP/IP protocol suite. SNMP runs over UDP, which in turn runs over IP.

Managed Objects: Values of network devices that can be read or overwritten by the SNMP manager, like alarm status, control relay status, system uptime, etc. In SNMP terms, every network device is defined in the MIB as a set of managed objects.

Management Information Base (MIB): The MIB is a data structure that describes SNMP network elements as a list of data objects. To monitor SNMP devices, your SNMP manager must compile the MIB file for each equipment type in your network.

Manager: A top-level SNMP master system (hardware or software) serving as the human interface to the SNMP network. The manager can issue Get, GetNext and Set requests to agents and receives GetResponse and Trap messages.

NMS: Network Management Software or Network Management System. Another term for SNMP manager software or hardware.

Object Identifier (OID): A number that uniquely identifies a managed object in an SNMP network. An OID consists of a series of numbers separated by decimal points. Each decimal point represents a leaf node in the tree structure of the MIB. For example, all OIDs for DPS Telecom equipment begin with the numbers 1.3.6.1.4.1.2682. This sequence represents: iso (1); org (3); dod (6); internet (1); private (4); enterprises (1); dpsInc (2682).

Ports 161 and 162: The virtual ports most commonly used to transmit SNMP messages. Port 161 is used for messages sent by the manager, and Port 162 carries messages sent in the opposite direction from agents.

Protocol Data Unit (PDU): An SNMP message. There are 5 types of PDU in SNMP v1: Get, GetNext, Set, GetResponse and Trap.

Protocol Data Unit (PDU): An SNMP message. There are 5 types of PDU in SNMP v1: Get, GetNext, Set, GetResponse and Trap.

Proxy agent: An SNMP agent that translates non-SNMP messages and inputs to SNMP. In network alarm monitoring, a proxy agent is usually an RTU that converts contact closure inputs to SNMP traps, like the NetGuardian 832A. Devices that mediate other alarms in other protocols to SNMP, like the NetMediator T2S (TBOS to SNMP) is also a proxy agent.

Referenced (RFC) MIBs: MIBs that are required by the main MIB during compiling. If any of these referenced MIBs are missing, the main MIB will not compile properly.

Set: An SNMP message issued by a manager instructing an agent to change a Managed object to a new value.

Simple Network Management Protocol (SNMP): the standard TCP/IP protocol for managing IP network devices.

Structure of Management Information (SMI): the standard that defines the MIB structure.

Standing alarm list: A list of all uncleared alarms, as maintained by a full-featured network alarm management system. Standard SNMP managers automatically delete all acknowledged traps, but a standing alarm list displays every alarm that has not been reported as cleared by the monitoring equipment.

Transmission Control Protocol (TCP): the more common transport layer protocol in the TCP/IP suite. TCP is considered a “reliable” protocol because it establishes a connection between the host and the recipient, guaranteeing delivery. UDP, the transport protocol used for SNMP does not establish a connection or guarantee delivery.

Trap: An SNMP message issued by an SNMP agent that reports an event.

User Datagram Protocol (UDP): the transport layer protocol used to send SNMP messages. Unlike TCP, UDP is a connectionless protocol that does not guarantee delivery of the data packet. However, UDP uses fewer network resources than TCP, making it more suitable for transporting a large number of status messages.

Variable Binding: the data field of a GetResponse or Trap PDU. Each variable binding lists a managed object and its current value.

Get the Facts Before You Purchase Your Next Network Monitoring System

If you found the information in this white paper useful, you'll also be interested in the other white papers in the DPS Telecom Network Monitoring Guide series. Each paper is a complete guide to an essential aspect of network monitoring. These are the facts you need to know to make an informed purchase of your next network monitoring system.

The 3 Fatal Mistakes Telecom Executives Commonly Make When They Attempt To Maintain Service Levels at Remote Sites In the Face Of Reduced Staffing ... And How You Can Avoid Them

Your network monitoring can be an asset to your business, or it can be a threat. Here are the three fatal mistakes telecom executives make in planning their network monitoring-and how you can avoid the mistakes and gain a competitive edge. To receive this report, send an e-mail to: 3fatalmistakes@dpstelecom.com.

SNMP Tutorial: A Fast Track Introduction to SNMP and its Practical Use in Network Alarm Management

An introduction to SNMP from the perspective of network alarm management. It summarizes the history and structure of the protocol, and offers some concrete applications for using SNMP for network alarm management. To receive this report, send an e-mail to: snmpfasttrack@dpstelecom.com.

Unsupported Legacy Network Alarm Monitoring Equipment: Why It's a Problem - What You Can Do About It

Many companies are dependent on legacy network monitoring equipment that is no longer supported by the manufacturer. This guide to legacy support issues explains why legacy equipment is a dead-end-and how you can escape the legacy trap. To receive this report, send an e-mail to: legacytrap@dpstelecom.com.

Give Us Your Feedback

Send your comments to feedback@dpstelecom.com

This all sounds great, but where can I get product details?

If you would like to know more about the products and services mentioned in this white paper, visit www.dpstelecom.com and click "Applications." or "Products."

"I would personally like to let you know how beneficial the installation of the SNMP responder was to the mission of our department. We were looking for a way to integrate our local ILEC region in HP OpenView without a major network change. The SNMP responder was the answer. This migration will allow us not only to monitor all alarms in one spot but also build extensive collection reports of our whole network."

—**Todd Matherne**, EATEL

"It is hard to find companies with the intelligence and aptitude to meet the customer's exact needs, and I believe that is what DPS is all about."

—**Lee Wells**, Pathnet

Written by Marshall DenHartog and Andrew Erickson

About the Author

Marshall DenHartog has ten years' experience working with SNMP, including designing private MIB extensions, creating SNMP systems for multiple platforms, and developing SNMP-based monitoring for several nationwide networks.

DenHartog's experience with both the theoretical and practical sides of SNMP have equipped him to write a straightforward guide to troubleshooting real-world SNMP problems.

US $36.95