

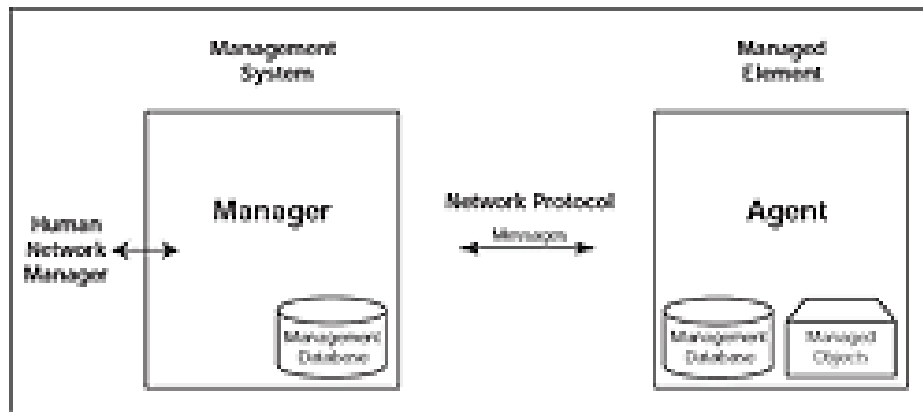
# SNMP Tutorial

(DPS Telecom)

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 Based on Manager/Agent Model

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 is based on the manager/agent model of a network management architecture.**

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.

## Five SNMP Command Messages

SNMP uses five basic messages (GET, GET-NEXT, GET-RESPONSE, SET, and TRAP) to communicate between the manager and the agent. The GET and GET-NEXT messages allow the manager to request information for a specific variable. The agent, upon receiving a GET or GET-NEXT message, will issue a GET-RESPONSE 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 in the case of an alarm remote that will operate a relay. The agent will then respond with a GET-RESPONSE 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.

As you can see, most of the messages (GET, GET-NEXT, and SET) are only issued by the SNMP manager. Because the TRAP message is the only message capable of being initiated by an agent, it is the message used by DPS Remote Telemetry Units (RTUs) to report alarms. This notifies the SNMP manager as soon as an alarm condition occurs, instead of waiting for the SNMP manager to ask.

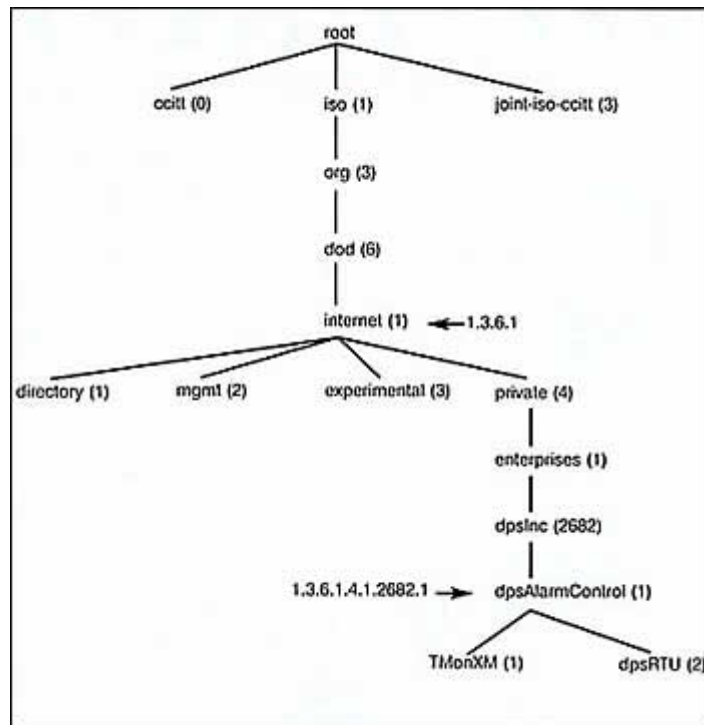
# SNMP Tutorial

(DPS Telecom)

## Simplicity of SNMP Leads to Widespread Use

The small number of commands used is only one of the reasons SNMP is "simple." The other simplifying factor is its reliance on an unsupervised or connectionless communication link. This simplicity has led directly to its widespread use, specifically in the Internet Network Management Framework. Within this framework, it is considered "robust" because of the independence of the managers from the agents, e.g. if an agent fails, the manager will continue to function, or vice versa. The unsupervised communication link does however create some interesting issues for network alarm monitoring we will discuss more thoroughly in a later issue of our tutorial.

Each SNMP element manages specific objects with each object having specific characteristics. Each object / characteristic has a unique object identifier (OID) consisting of numbers separated by decimal points (i.e., 1.3.6.1.4.1.2682.1). These object identifiers naturally form a tree as shown below. The SNMP MIB associates each OID with a readable label (i.e., dpsRTUASState) and various other parameters related to the object. The MIB then serves as a data dictionary or code book that is used to assemble and interpret SNMP messages.



The branch of the MIB object identifier tree.

## SNMP Packets Require OIDs

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 SNMP 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.

When an element sends an SNMP TRAP packet, it can include OID and value information (bindings) to clarify the event. DPS remote units send a comprehensive set of bindings with each TRAP to maintain traditional telemetry event visibility. Well-designed SNMP managers can use the bindings to

# SNMP Tutorial

## (DPS Telecom)

correlate and manage the events. SNMP managers will also generally display the readable labels to facilitate user understanding and decision-making.

This part of our tutorial on the Simple Network Management Protocol (SNMP) examines the communication between SNMP managers and SNMP agents. Basic serial telemetry protocols, like TBOS, are byte oriented with a single byte exchanged to communicate. Expanded serial telemetry protocols, like TABS, are packet oriented with packets of bytes exchanged to communicate. The packets contain header, data and checksum bytes. SNMP is also packet oriented with the following SNMP v1 packets (Protocol Data Units or PDUs) used to communicate:

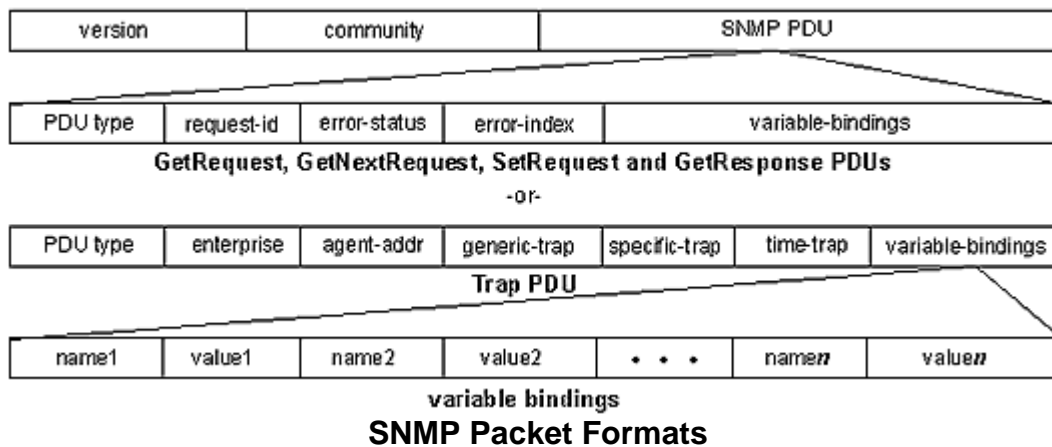
- Get
- GetNext
- Set
- Trap

### Set Requests Change Variables Within Managed SNMP Devices

The SNMP manager sends a Get or GetNext to read a variable or variables and the agent's response contains the requested information if managed. The manager sends a Set to change a variable or variables and the agent's response confirms the change if allowed. The agent sends a Trap when a specific event occurs.

The image below shows the SNMP packet formats. Each variable binding contains an identifier, a type and a value (if a Set or response). The agent checks each identifier against its MIB to determine whether the object is managed and changeable (if processing a Set). The manager uses its MIB to display the readable name of the variable and sometimes interpret its value.

In this fourth article in our series, we continue to examine the Simple Network Management Protocol (SNMP) focusing specifically on the layered communication model used to exchange information. Our last article focused on the structure of SNMP messages, however an SNMP message is not sent by itself. It is wrapped in the User Datagram Protocol (UDP), which in turn is wrapped in the Internet Protocol (IP). These are commonly referred to as layers and are based on a four-layer model developed by the Department of Defense (you may recall the DoD origins of the Internet).



### SNMP Resides on Application Layer

SNMP resides in what is called the Application layer, UDP resides in the Transport layer and IP resides in the Internet layer (somewhat obvious). The fourth layer is the Network Interface layer where

# SNMP Tutorial

(DPS Telecom)

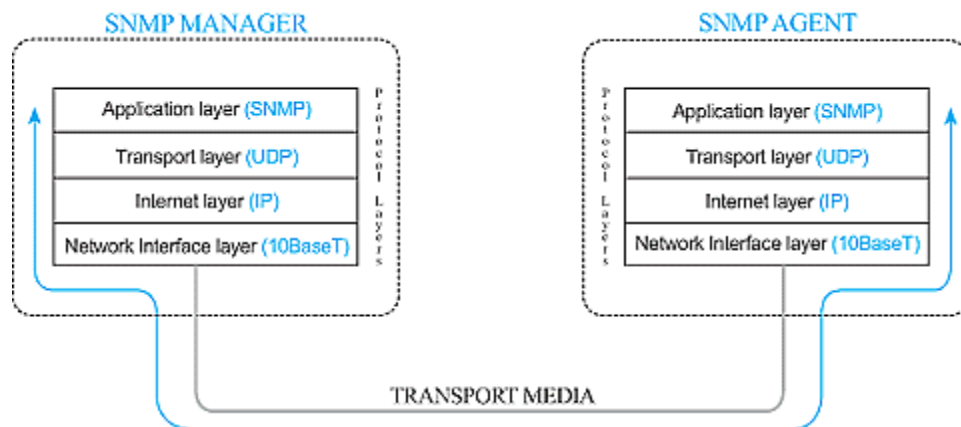
the assembled packet is actually interfaced to some kind of transport media (i.e., twisted pair copper, RG58 co-axial or fiber). While this multi-layer model may seem a bit confusing, it effectively isolates the tasks of communication and ultimately assists in designing and implementing a network.

## Traversing the Layers

To illustrate the function of this layered model, let's look at a single SNMP GET request from the agent's perspective. The SNMP manager wants to know what the Agent's System Name is and prepares a GET message for the appropriate OID. It then passes the message to the UDP layer. The UDP layer adds a data block that identifies the manager port to which the response packet should be sent and the port on which it expects the SNMP agent to be listening for messages. The packet thus formed is then passed to the IP layer. Here a data block containing the IP and Media Access addresses of the manager and the agent is added before the entire assembled packet gets passed to the Network Interface layer. The Network Interface layer verifies media access and availability and places the packet on the media for transport.

## Agents Pull SNMP Traps from the Network Interface Layer

After working its way across bridges and through routers (the modern equivalent of over the rivers and through the woods) based on the IP information, the packet finally arrives at the SNMP agent. Here it passes through the same four layers in exactly the opposite order as it did at the SNMP manager. First, it is pulled off the media by the Network Interface layer. After confirming that the packet is intact and valid, the Network Interface layer simply passes it to the IP layer. The IP layer verifies the Media Access and IP address and passes it on to the UDP layer where the target port is checked for connected applications. If an application is listening at the target port, the packet is passed to the Application layer. If the listening application is the SNMP agent, the GET request is processed as we have discussed in previous articles. The agent response then follows the identical path in reverse to reach the manager.



**An SNMP message passes through the protocol layers at both the manager and the agent. Each layer addresses a specific communication task.**

## An Aid for Troubleshooting

Understanding this layered model of SNMP communications makes it easier to troubleshoot network problems. When there is a problem, you can simply trace it down, out one end, into, and up the other. LAN/WAN link and activity status indicators provide some visibility to the Network Interface layer.

ICMP echo requests and responses (Pings) provide some information regarding the proper functioning of the IP layer. SNMP processing indicators can be used to verify the passage of the

# SNMP Tutorial

## (DPS Telecom)

SNMP packet through the UDP layer and the functioning of the Application layer. Each step can be verified independently until all steps are working correctly for end-to-end communication.

SNMP is a standard protocol that has wide acceptance in the industry and is flexible enough to describe almost anything. Because of these advantages, many network managers have come to believe that SNMP should be used for all network monitoring applications.

### **Not All SNMP Managers Can Provide Network Visibility and Control**

SNMP certainly has its place in an effective telecom network management solution, but this doesn't mean that any off-the-shelf SNMP manager can provide adequate visibility and control of your network.

### **Off-The-Shelf SNMP Solutions Cannot Meet Real-World Needs**

The typical off-the-shelf SNMP manager is not designed for displaying and processing telemetry data for effective network monitoring, especially for the kind of real-world monitoring tasks network managers most need performed. These capabilities can be added to an SNMP manager, but it usually requires substantial custom software development.

### **Make sure you avoid these 7 common mistakes**

Relying on off-the-shelf SNMP systems for mission-critical telemetry is a major mistake. If you're switching from traditional telemetry or integrating non-SNMP monitoring with an SNMP-based system, an off-the-shelf SNMP manager will not provide the detailed alarm data you expect. Before you commit to an SNMP monitoring solution, you need to make sure it supports essential network alarm monitoring functions.

There are seven common mistakes network managers typically make when integrating SNMP and non-SNMP monitoring. Your SNMP implementation will be successful only if you can avoid them.

#### **1. Selecting an SNMP system that doesn't provide complete, precise alarm descriptions**

A basic SNMP manager doesn't record the location, time, severity, or a precise description of alarm events. To adapt an off-the-shelf SNMP manager to monitor these factors, you must create and maintain a master alarm list representing all the monitored points in your network - and then also create and maintain a database associating all the traps that may be sent to the SNMP manager with the alarms on that list.

#### **2. Settling for an SNMP system that can't identify cleared alarms**

Even more database work is required to identify whether a trap corresponds to an alarm condition or a clear condition. Creating this addition to the trap association database often requires analyzing multiple variable bindings within the trap packet.

#### **3. Not maintaining a history of standing SNMP alarms**

Relying solely on a basic SNMP manager for network alarm monitoring can potentially result in completely losing visibility of threats to your network. A basic SNMP manager doesn't maintain a list of standing alarms. Instead, the typical SNMP manager maintains an event log of newly reported traps and a history log of acknowledged traps. As soon as a trap is acknowledged, it is considered cleared. Imagine what might happen to your network if a system operator acknowledges an alarm, and then, for whatever reason, fails to correct the alarm condition. Who would know the alarm is still standing?

#### **4. Not identifying SNMP system operators**

Basic SNMP managers do not record the identity of the system operator who acknowledges an

# SNMP Tutorial

(DPS Telecom)

alarm. In the example of the negligent system operator, it would be impossible to determine who had made the mistake or to assign responsibility for the resulting problems.

## 5. **Trusting an SNMP system that's insecure for multiple users**

Out of the box, the typical SNMP manager is not designed for multi-user security. All traps are posted to one alarm list; all users may view all alarms, and all users may acknowledge all alarms.

## 6. **Broadcasting all SNMP alarms to all system users**

Basic SNMP managers have no built-in functions for organizing alarms by logical category, posting the same alarm to multiple logical categories, or sorting which alarms the user wants to see. If Jones is in charge of all equipment for the Western region, and Smith is in charge of power plants, both need to know about a generator failure in Tucson, but neither one needs to know about all the alarms in the network. And if one manager corrects the alarm condition and acknowledges the alarm, the other manager needs to know it was acknowledged and by whom. Unfortunately, standard SNMP managers will not support these functions.

## 7. **Allowing yourself to be bombarded by nuisance SNMP alarms**

No SNMP manager supports the advanced features necessary for best quality telemetry monitoring, such as notifications escalation, legacy protocol mediation, nuisance alarm silencing, automatic control relay operation, and automatic notifications by pager and e-mail.

## **Requirements for Extensive Customization Reduce the Advantages of an Open Standard**

It is true that many, but not all, of these functions can be added to standard SNMP managers, but implementing network alarm monitoring in a basic SNMP manager usually involves a substantial amount of custom software module development. Even when pre-built software modules are available, they usually require custom tweaking to perform exactly as you want them to.

The need for extensive customization eliminates the advantage of using a simple open standard, and it is difficult to justify significant development costs after purchasing an already expensive SNMP manager. Why take the time, trouble, and expense to recreate capabilities that are already present in a high-quality, SNMP-capable network alarm management system?

## **The Right Role for Your SNMP manager**

Relying on an SNMP manager for critical network monitoring just doesn't take into account the tons of legacy and non-SNMP equipment that is functioning perfectly fine out in networks all over the world. The role of an SNMP manager is best used for inventorying network devices and drilling down into equipment details after your network monitoring system notifies you of a problem.

SNMP can be an effective tool, but it's only one item in your network alarm monitoring toolkit, and it can be used more effectively when it is part of a total network monitoring solution.