TECH TIP – WHAT IS SNMP?

Subject: What is SNMP? **Date: 17th March 2014**

Description

Simple Network Management Protocol (commonly referred to as "SNMP") is a popular protocol used for network management. It is widely used for collecting information from, and configuring, devices such as servers, hubs, switches, routers on an Internet protocol (IP) network.

In more recent times, and driven by the availability of IP connectivity between equipment locations, SNMP has become more widely used to interact with other types of devices - such as solar chargers, UPS power supplies, two-way radio base stations, trunk network site switches and other wireless infrastructure equipment (such as RFI's Antenna System Monitor or ASM). SNMP provides an independent and standards-defined communications protocol that allows multiple devices, from multiple suppliers, to communicate openly.

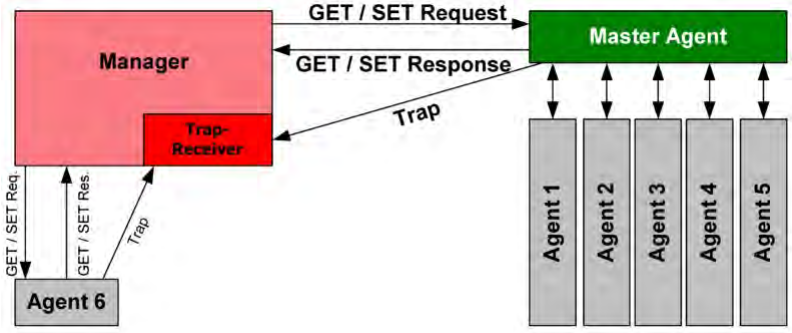
Several versions of the SNMP protocol exist. The earliest implementation was called SNMPv1, a later release was SNMPv2 (with revision "C" of SNMPv2 being widely used), and the latest release is SNMPv3 (which uses encryption to prevent unauthorised use of SNMP messages).

History

Developed in 1988 to provide network device monitoring capability for TCP/IP-based networks, SNMP was approved as an Internet standard in 1990 by the Internet Architecture Board (IAB) and has been in widespread use ever since…..

Terminology

In typical SNMP uses, one (or more) administrative computers, called a *Manager*, has the task of monitoring or managing a group of devices connected via a TCP/IP network. Each device being monitored runs some software called an *Agent* which reports information from that device to the *Manager(s)*. In networks with a large number of devices, a *Master Agent* can also be implemented to amalgamate clusters of Agents – reducing the number of individual Agents (or devices) being managed directly by a Manager.



Typical SNMP network architecture

SNMP Agents expose management data for the devices as *variables*. In some cases text "names" can also be included in the SNMP messages sent by *Agents*. The SNMP protocol also permits device management tasks, such as modifying and applying new configuration data, through modification of these variables.

SNMP itself does not define what information (i.e. which variables, and how many of them) a device should offer to a Manager. Rather, these variables are organised in specific hierarchies, which are described by *Management Information Bases* (or "MIBs"). These are commonly just referred to as "MIB files".

SNMP Message Structure

So, what does an SNMP message look like? It's a text message that follows the rules of the referenced MIB's heirarchy.

The message at right is an example extract of an Antenna System Monitor (ASM) SNMP "Trap" alarm message. It contains *Object Identifiers* OIDs and, for variables that can have supporting text content, the text itself (i.e. "Mt Dandenong").

```
11:14:21 AM  18/03/2014
Received From: 202.87.8.210  Mt Dandenong UHF
Version: 2  Community: public  Type: &HA7
Std PDU [RequestId: 0, Length: 387, ErrStatus:0, ErrIndex: 0]

1.3.6.1.2.1.1.3.0 6193800
1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.4.1.380.71.2.2.2.1.6.0.1
1.3.6.1.4.1.380.71.2.2.2.1.2.1.0 Demo
1.3.6.1.4.1.380.71.2.2.2.1.2.2.0 Mt Dandenong
1.3.6.1.4.1.380.71.2.2.2.1.2.3.0 2
1.3.6.1.4.1.380.71.2.2.2.1.2.4.0 160
1.3.6.1.4.1.380.71.2.2.2.1.2.5.0 PWR=OK(-108.7); VCO=OK; RX=ON
1.3.6.1.4.1.380.71.2.2.2.1.2.6.0 1
1.3.6.1.4.1.380.71.2.2.2.1.2.7.0 •Þ└↕♪ß┬
```

Example extract from a SNMP Trap message

MIB Files

MIBs are like an index, and describe the structure of the available management data of each device within the heirarchy, using numeric labels called *Object Identifiers* ("OIDs"). Each OID represents a variable in the device that can be read or set via SNMP.

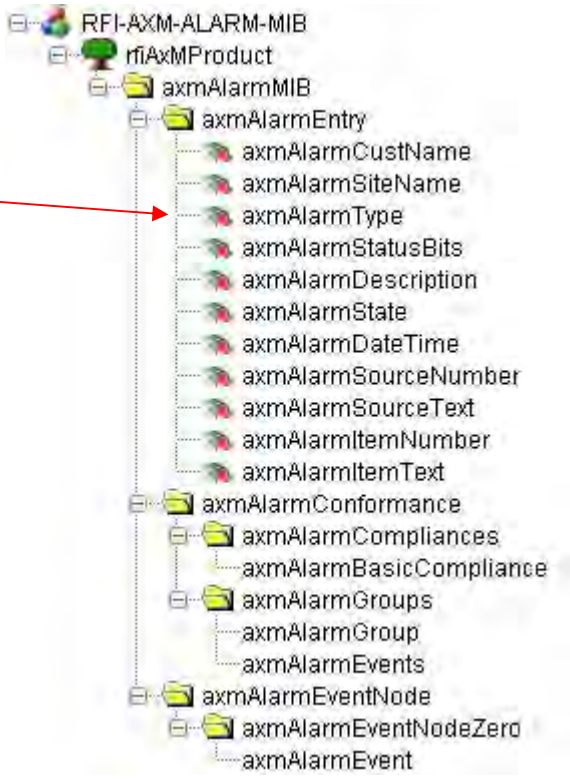
Each portion of the SNMP message can be translated to a position in the MIB structure, with a specific format and content expected within the identified heirarchy.

For example, if we look at *axmAlarmType* in the MIB shown at right, the OID for this is …..2.2.2.1.2.3 and the defined values in the MIB for this OID are as follows;

Description	MultiVar
Syntax	INTEGER { Restart (0) , SystemStatus (1) , ChannelStatus (2) , CamStatus (3) }
Access	read-only
Index	
Object ID	.1.3.6.1.4.1.32327.2.2.2.1.2.3
Description	"Number identifying the type of alarm."

Definition of the axmAlarmType message in the MIB

So in the Antenna System Monitor's MIB, the Alarm Type can be one of 4 values; a "0" (Restart), a "1" (System Status), a "2" (Channel Status), or a "3" (CamStatus). In other words, when the *Manager* receives this SNMP message and interprets each byte of data within it, it can look up the value received in this position of the message and know it is a Restart, System Status, Channel Status or CAM Module Status message.



The Antenna System Monitor's MIB heirarchy

Decoding the SNMP Message

For interest, let's analyse part of an SNMP Trap message….

Refering to the axmAlarmType message Object ID (OID) shown above (…….2.2.2.1.2.3) and locating it within the example SNMP Trap message, we can see that the variable value for this OID is a "2" with (……2.2.2.1.2.3.0.2).

We could then look at another portion of the SNMP Trap message and the MIB would help us define that too…..

axmAlarmDescription message Object ID (OID) shown (…….2.2.2.1.2.5) requires the textual description for the Alarm Type "PWR=OK(-108.7); VCO=OK; RX=ON"

```
11:14:21 AM  18/03/2014
Received From: 202.87.8.210  Mt Dandenong UHF
Version: 2  Community: public  Type: &HA7
Std PDU [RequestId: 0, Length: 387, ErrStatus:0, ErrIndex: 0]

1.3.6.1.2.1.1.3.0 6193800
1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.4.1.380.71.2.2.2.1.6.0.1
1.3.6.1.4.1.380.71.2.2.2.1.2.1.0 Demo
1.3.6.1.4.1.380.71.2.2.2.1.2.2.0 Mt Dandenong
1.3.6.1.4.1.380.71.2.2.2.1.2.3.0 2
1.3.6.1.4.1.380.71.2.2.2.1.2.4.0 160
1.3.6.1.4.1.380.71.2.2.2.1.2.5.0 PWR=OK(-108.7); VCO=OK; RX=ON
1.3.6.1.4.1.380.71.2.2.2.1.2.6.0 1
1.3.6.1.4.1.380.71.2.2.2.1.2.7.0
```

Example extract from a SNMP Trap message

Description	MultiVar
Syntax	DisplayString
Access	read-only
Index	
Object ID	.1.3.6.1.4.1.32327.2.2.2.1.2.5
Description	"Textual description of the alarm."

So the *Manager* would be able to interpret from the MIB that this as a Channel Status Message, with Rx PWR being reported as OK at -108.7(dBm), the ASM's VCO is OK and the Rx Carrier is present (ON).

We could use the RFI ASM MIB file to further decode the rest of the SNMP Trap message if we wished……but the *Manager* does that anyway.

Communications Directions

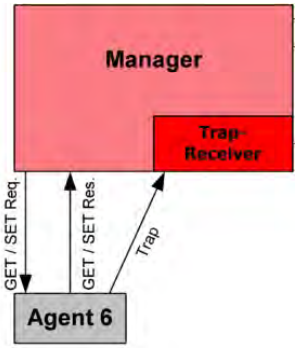
Communications between the Manager and other devices in the architecture are often referred to by the direction of their communications.

Southbound From the *Manager* "down" to an *Agent* under its management, and,

Northbound From an *Agent* "up" to its *Manager*

In the diagram shown at right, it can be seen that a *Trap* message would be considered a *northbound* message. This is why we say "the ASM can provide SNMP northbound traps".

Similarly, a *GetRequest* is considered a *southbound* message, and the *Response* to that *GetRequest* would be a *northbound* message.



SNMP Message Types

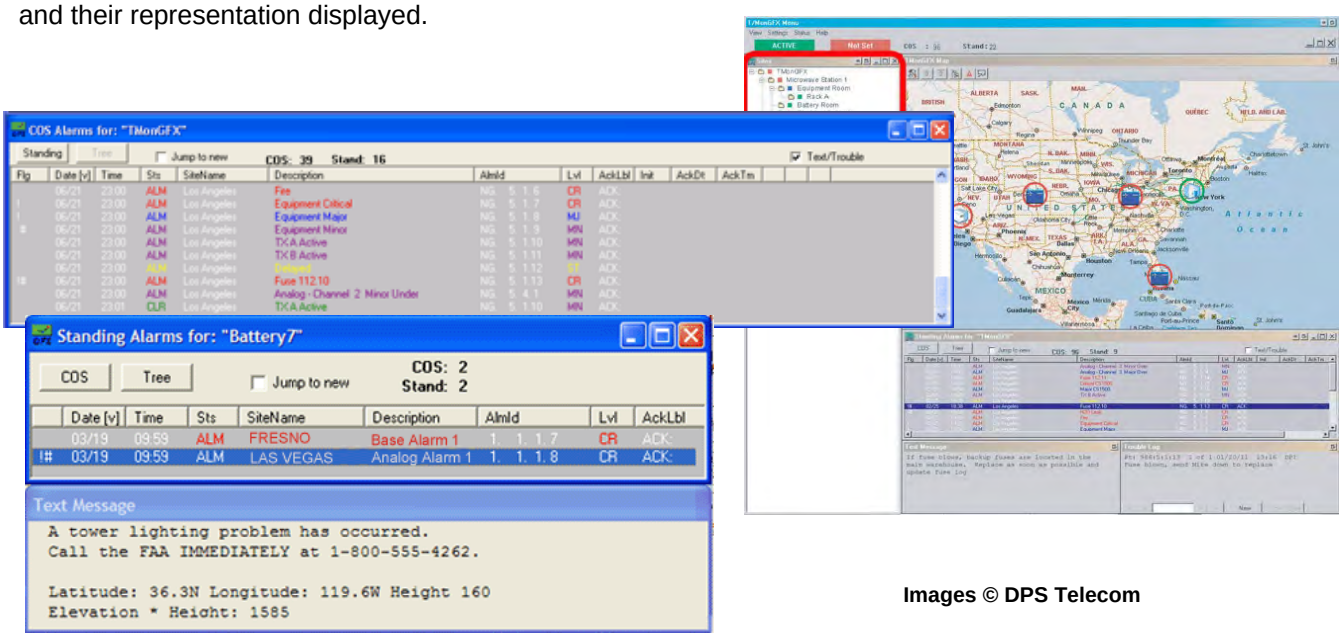
We have only looked at one example of an SNMP message - a “Trap”. In SNMPv1 there are five (5) message types, and in SNMPv2 there are seven (7). SNMPv1 and SNMPv2 messages are sent as plain text. SNMPv3 encrypts these text messages so they cannot be seen (or “sniffed”) on a network and used by an unauthorised party.

These messages types are;

GetRequest	A Manager-to-Agent request to retrieve the value of a variable or a list of variables. A *Response* with the variable’s current value(s) is returned.
SetRequest	A Manager-to-Agent request to change the value of a variable or a list of variables. A *Response* with the variable’s new value(s) is returned.
GetNextRequest	A Manager-to-Agent request to discover available variables and their values. A *Response* with the next variable in the MIB is returned.
GetBulkRequest	An optimised version of *GetNextRequest*. A Manager-to-Agent request for multiple iterations of the *GetNextRequest*. Returns a Response containing multiple variables. This message type was introduced in SNMPv2.
Response	Returns variables and acknowledgements from Agent-to-Manager for the above message types. This message type was known as *GetResponse* in SNMPv1.
Trap	A notification from Agent-to-Manager. SNMP traps enable an Agent to notify the Manager of significant events (such as alarms) by way of an unsolicited message.
InformRequest	An acknowledged notification. An unsolicited Trap message is answered by the Manager upon receipt, so an Agent knows its Trap has been received. This message type was introduced in SNMPv2.

Managing a Network of Agents (Devices)

It can be appreciated that a software application running on a *Manager* could monitor and manage a whole network of different *Agents* (devices), and that by having the MIB file relevant to each type of device to use as an index, the SNMP messages received from each type of device could be interpreted and their representation displayed.



Images © DPS Telecom

Acknowledgements:

Some portions of this document are subject to the GNU Free Documentation Licence and are reproduced by permission.
All rights remain with their originator.

- END -