SNMP – a new paradigm for SCADA

By Vishal Prakash and Robert Casey
Introduction

Remote Monitoring and diagnostics of industrial devices has evolved significantly over the years. In the seventies and early eighties remote monitoring was mainly via 1200 baud modems which was considered fast (not to mention expensive!). The 1200 baud links were considered fast, sufficient and reliable because of the nature of many of the telemetry protocols which are typically deterministic by nature. We say deterministic because of the way these protocols handled the assertion of monitoring and control messages, feedback messages of various types and relay (store & forward) messages. This transparency and flexibility gave telemetry protocols the edge over other industrial protocols which in many cases were hard-wired. A good example of a deterministic telemetry protocol is Modbus.

During this same period (70’s, 80’s and early 90’s) the internet was virtually unknown and Ethernet had just been invented. However, the outburst of Internet and the falling cost of IP-based hardware means that the number of IP based devices that are used for industrial control and monitoring has skyrocketed. Most of these devices support their own proprietary protocol. Proprietary protocols are rigid to implement in a remote telemetry system and even harder to manage from a SCADA perspective.

The explosion in the use of IP-based remote control and monitoring equipment secured the need to develop a common interface or protocol which could overcome the heterogeneity of these modern industrial telecontrol and network devices. Unification of the management information and of the administration protocol became compulsory which could be used to remotely monitor and diagnose these devices. The protocol that was developed for this purpose is called SNMP which stands for Simple Network Management Protocol.

History

SNMP was developed by a team of six people in 1988. It is an application-layer protocol which has achieved widespread acceptance as a platform for the monitoring and management of network attached devices. This management protocol, derived from a predecessor management protocol, Simple Gateway Management Protocol (SGMP), was initially developed as a short-term solution, intended to be replaced by a solution based on the Common Management Information Service and Protocol (CMIS/CMIP) architecture, although this later solution never gained the widespread acceptance of SNMP.

With this increasing acceptance and adoption, SNMP is now used across a wide variety of platforms and network environments, in particular, information technology and broadcast where SNMP is used as a unified mechanism to monitor the status and operation of a diverse range of equipment.

Protocol Definition

SNMP or Simple Network Management Protocol is an application layer asymmetric protocol which has been widely accepted by the IT community as a platform for the monitoring and management of a wide range of network attached devices. The communications is generally between a management station (top-end) and an agent (remote device).
How does SNMP Protocol work?

In broad terms, SNMP as a management model consists of:

- At least one, but generally many, devices or nodes that contain an SNMP application component that exposes elements of its configuration or operation for management through the SNMP protocol – These nodes are traditionally referred to as agents and comprise those elements of the network subject to monitoring and management using SNMP. From a more traditional SCADA perspective, these agents could be referred to as SNMP slave devices.

- At least one, but potentially more, SNMP node that is capable of initiating management commands and or receiving monitoring notifications from other devices – A node of this manner is traditionally referred to as a manager. From a more traditional SCADA perspective, these agents could be referred to as SNMP master devices.

This architectural model in turn permits the management and monitoring of network attached devices by the direct polling and query of an SNMP agent by a manager using a relatively small set of commands, or the asynchronous receipt of a monitoring notification, referred to as a trap (or as an exception report in more traditional SCADA terms) by an SNMP manager from an agent.

The relationship between the elements of the SNMP architectural model may be represented diagrammatically as follows.

![SNMP Architecture Diagram](image)

The configuration or operational elements that are exposed to monitoring or management through the SNMP protocol are typically referred to as objects – The list of these objects, typically referred to as a MIB (or management information base), available within a given SNMP device is highly variable and while there are some “standard” recommendations with respect to configuration items that should be made available for any given device, there is absolutely no mandate for this to be the case. As such, when interfacing to a new device using SNMP, the first level of information that is generally sought is the device MIB file that lists and describes the SNMP objects of the device.

**Management Information System**

MIB or Management Information System consists of a collection of individual objects or elements such as point status of instrumentation signals that can be monitored or controlled using the SNMP protocol. The SNMP manager and agent employ a common MIB. The individual objects also called as managed objects are organ-
ized within a MIB in a hierarchical structure that when laid out looks like a tree (see Figure 2). Each managed object has a unique OID or Object Identifier consisting of numbers separated by decimal points (example: 1.5.6.3.4.3981.2). The OID for each object is determined by the position of the managed object in the tree-like hierarchical structure. The OID is used to distinguish each object in a SNMP message.

**SNMP Messages**

- **GET and GET-NEXT** – these two messages are used the manager to obtain information from an agent like the status of I/O
- **GET-RESPONSE** is a message initiated by the agent to the manager upon receiving a GET or GET-NEXT message
- **SET** is a message used by the manager to set the value of a managed object within a MIB
- **TRAP** messages are generated asynchronously by agents containing alarm of status change information. TRAP messages are always sent from an agent to a manager.

![Hierarchical structure of a MIB in a SNMP agent](image)

**SNMP Manager**

![SNMP Agent — Semaphore's Kingfisher PLUS + RTU](image)
It should be noted that while a total of five messages can lead to the perception that SNMP is a simple protocol; the truth is far from it. The five messages do a good job of hiding the detail of interface and complexity of management operations that may be exposed through SNMP for a given device.

**SNMP in Semaphore’s Remote Terminal Units**

Semaphore has implemented the SNMP protocol in our Kingfisher PLUS+, G30 and MS-CPU32 RTUs. The protocol implementation is in three parts:

- SNMP Trap Protocol Implementation
- SNMP Daemon Protocol Implementation and
- SNMP Client Protocol Implementation

Semaphore’s Kingfisher PLUS+ RTU is a modular RTU with full redundancy, smart I/O modules, ISaGRAF and support for multiple communications media. The CPU is 32-bit ARM9 with Linux RTOS. Kingfisher PLUS+ RTUs support over 70 different protocols including SNMP, DNP3, AB DF1, User Defined and Modbus.

Semaphore’s Tbox MS-CPU32E is a modular RTU with support for integrated communication options like GSM/GPRS and full IP capability with free web server, FTP, e-mail and NTP. The CPU is a 32-bit Power PC CPU with Linux RTOS T-Box RTUs support over 40 different protocols including SNMP, Modbus and AB DF1.

Semaphore’s G30 RTU is an all-in-one RTU with PS, communications, CPU, I/O daughter cards and ISaGRAF. The CPU is 32-bit ARM9 with Linux RTOS. The G30 features a unique oLED display and support multiple protocols including SNMP, DNP3 and Modbus.

Figure 4: Semaphore RTUs that support SNMP protocol
**SNMP Messages**

The SNMP trap protocol implementation allows Semaphore RTUs to send and receive SNMP trap messages. In this manner, the RTUs may be used to monitor system health and alarm states for information technology or broadcast equipment where the trap functionality associated with SNMP is widely used to provide a mechanism to monitor equipment of different models and from different manufacturers in a unified manner. Alternatively, using the ability to generate SNMP traps, the RTUs may be integrated in such network environments alleviating the requirement for a separate supervisory system.

**SNMP Daemon Protocol**

The SNMP daemon protocol implementation provides a server (or slave) SNMP implementation which allows configuration and state information about RTUs to be set, queried or retrieved by remote agents using SNMP. This functionality permits the RTUs to be integrated within network environments where SNMP is employed and administered using existing SNMP management tools. The configuration and state information about the RTUs exposed through this interface are defined in the Semaphore MIB Implementation and include:

- RTU network address and system identifier;
- RTU hardware modules and I/O states;
- Event log information; and
- Network interface and traffic information.

The Semaphore MIB defines those elements of RTU hardware and software configuration available for query and manipulation through the SNMP agent interface provided by the SNMP daemon protocol implementation. The Semaphore RTUs also support the Management Information Base for Network Management of TCP/IP based internets (MIB-II) as defined in RFC document 1213 which describes SNMP objects that expose information about system configuration, network interfaces and connection and protocol information. The Semaphore MIB definition is available as a separate file to download.

**SNMP Client Protocol**

The SNMP client protocol implementation provides a client (or master) SNMP implementation that allows the RTUs to be used to query, retrieve and set information associated with remote devices using SNMP. In this manner this protocol implementation may be used to extend the scope of monitoring of the RTUs beyond traditional SCADA applications and physical I/O to incorporate the monitoring of SNMP enabled network equipment.

**SNMP Function Blocks**

Semaphore has developed function blocks which can be used with the implementation of the SNMP protocol as described above. The function blocks make it easier for the end user to implement SNMP in a telemetry system. When using the function, the user simply has to parse the correct parameters and the protocol handler does the rest!

The function blocks implemented by Semaphore include:

- **SNMP GET TRAP** – Used with SNMP Trap Protocol
- **SNMP SEND TRAP** - Used with SNMP Trap Protocol
- **SNMP GET INTEGER** – Used to retrieve a integer value associated with a managed object in a SNMP agent
SNMP – a new paradigm for SCADA

- SNMP GET UNSIGNED INTEGER – As above but used to retrieve unsigned integers
- SNMP GET STRING – Used to retrieve a string value associated with a managed object in a SNMP agent
- SNMP GET OBJECT IDENTIFIER – Used to retrieve the object identifier value associated with a managed object in a SNMP agent
- SNMP SET INTEGER – Used by a SNMP Manager to set an integer value associated with a managed object in a SNMP agent
- SNMP SET UNSIGNED INTEGER – As above except to set unsigned integer values
- SNMP SET STRING – Used to set a string value associated with a managed object in a SNMP agent
- SNMP SET OBJECT IDENTIFIER – Used to set the object identifier for a managed object in a SNMP agent

Examples of function blocks available for SNMP implementation in Semaphore RTUs. The function blocks make it easier for the end user to implement SNMP. Any condition can be used to trigger the function blocks and only the appropriate parameters will have to be parsed to the function block.

Figure 5: Example Function Blocks available in Semaphore RTUs for SNMP implementation

Benefits of SNMP (What, Why and How much?)

Before introducing SNMP to a remote monitoring system, one will have to answer the following questions at a minimum.

- What are the associated costs?
- What implications (like loss of functionality) does it have on the existing system?
- What are the benefits with respect to the current system?
- Why change when the system has been working as is forever?

There is no doubt that the above mentioned questions will have to be answered effectively prior to SNMP implementation in a remote monitoring system.

First and foremost a good understanding of SNMP protocol is required. There is a wealth of information available on this topic on the web including tutorials. Google-ing the word SNMP will produce 1000’s of hits! We also hope to have provided the readers of this white paper a basic understanding of SNMP and its implementation in Semaphore Remote Terminal Units.

A typical remote site has multiple types of hardware – RTU, network devices and other smart industrial devices. It can be reasonably assumed that all of these devices have their own protocols. As such the top end will have to interrogate the devices using the different protocols, one at a time. The top-end will then have to manage the different types of data received from the various devices. This will lead to an overall decrease in the efficiency of the remote monitoring and control system thereby defeating the purpose of such a system. Today, such quagmire systems exist and work at an acceptable level to the satisfaction of owner of the system.
So, why change? What will be the unique value proposition in adding SNMP to the already crowded system? The unique value proposition that SNMP brings to an existing telemetry monitoring and control system is that it can be used to provide control and monitoring of existing network based devices with access to physical I/O in a manner that would a natural extension of the existing system.

The structure of SNMP protocol makes it ideal for telemetry based control and monitoring systems. The protocol is assertive and handles the control, monitoring, feedback and acknowledgement messages efficiently. SNMP is also very transparent and flexible in its structure. SNMP is also inherently built into many of the network devices that are commonly present at a typical remote site. Thus, the incorporation of SNMP in Semaphore RTU’s means these network based devices can be diagnosed remotely without overloading the top-end SCADA or the communications infrastructure.

Semaphore’s implementation of SNMP is such that it will work on existing telemetry networks and communication infrastructure. The use of smart function blocks to access SNMP functionality and then storing results in variables mean the existing top-end design does not have to be changed. This means the costs of implementing SNMP in existing Semaphore telemetry systems is minimal. For new Semaphore customers the implication costs of SNMP is also minimal as our RTUs can communicate SNMP locally and then communicate to the top-end with other open protocols like Modbus and DNP3.

**Conclusion**

The primary benefit that SNMP offers end-users of Semaphore RTU technology is that it opens new opportunities to extend the reach of monitoring and control – Through the use of SNMP, the Semaphore RTU can query, monitor and manage the status and operation of SNMP devices, removing the requirement for multiple monitoring systems within IP networks, where deterministic control is required for business processes. Moreover, Semaphore RTUs using SNMP can extend the reach of existing SNMP managed networks providing access to physical contacts and I/O states in a manner which is a natural extension to the existing operation of the network.

In short the inclusion of SNMP in Semaphore RTUs provides a strong value proposition for the extension and addition of existing telemetry and IP networks, simplifying and removing the duplication that can exist in device monitoring and management between these realms.

**About Semaphore**

Semaphore offers the first IP-based RTU solutions that enable complete integration of SCADA, control, and communications functionality in one rugged package. Our simple yet powerful products leverage easy-to-use Web technologies and inexpensive public networks. They are easy to configure and offer dramatically reduced costs versus traditional SCADA/PLC systems.

Semaphore is a part of CSE-Global, a leading systems integrator with an international presence spanning the Americas, Asia Pacific, Europe, Africa, and the Middle East. The group employs over 1,200 people worldwide, with more than 85% representing design, engineering, and project management capabilities and experience. That makes CSE one of the largest independent system providers of its kind.