Network Management of Real-Time Embedded Processors

C. Kalbfleisch, S. Hunt, K. Low, D. Mathieson
SSC Laboratory *
2550 Beckleymeade Ave.
Dallas, Texas 75237

Abstract

The Superconducting Super Collider Laboratory is a complex of particle accelerators being built in Ellis County, Texas. It will have a dedicated global communications network that will deliver control messages and provide for general data acquisition. This network will connect thousands of computer nodes over a very large geographic area. In order to meet the demanding availability requirements being levied on the system, it will need comprehensive network management. A large number of the computer nodes are embedded systems that traditionally do not support network management services. This presents unique challenges to standard network management practices. The Simple Network Management Protocol, SNMP, is widely accepted by industry as a tool to manage network devices. In this paper we will examine the performance characteristics and usefulness of an SNMP agent in a real-time environment.

I. Network Management Historical Perspective

The Internet Activities Board (IAB) has spent considerable time focusing on standards for network management. In 1987 a group of engineers implemented the Simple Gateway Management Protocol (SGMP). Around the same time, the OSI network management documents specified CMIP over TCP (CMOT). (CMIP is the OSI Common Management Information Protocol, and TCP is the Transmission Control Protocol). These two groups met to determine if a consensus could be reached on a network management approach. The result of the meeting was that the SGMP protocol would be extended to address the needs of network devices other than gateways. It would be called the Simple Network Management Protocol (SNMP). This was to become the short term solution for network management in the community while a second group worked on the OSI approach as a long term solution. A third group was to design a common framework for network management so as to make the migration from SNMP to CMOT easier. By the fall of 1989 a number of vendors had implementations of SNMP installed, and it became the de facto operational standard for network management of TCP/IP based networks[1].

II. Structure of Management Information

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC02-89ER40486.

The Structure of Management Information (SMI) was initially deployed in order that the SNMP and the CMOT camps would have a common framework for identifiying managed objects[2]. A collection of managed objects is referred to as a Management Information Base (MIB)[3]. Essentially the SMI specifies a syntax for defining MIBs in Abstrax Syntax Notation One (ASN.1) macros and a base group of object types. Complex object types can be created using ASN.1. Additionally, a set of Basic Encoding Rules (BER) are defined to translate the ASN.1 instances into serialized octet strings that can be sent out onto a network.

Objects are defined with an associated Object Identifier (OID) in a tree structure. There are four branches in the tree that are of primary interest; Directory, Management, Experimental and Private. The Directory subtree is reserved for future use with the OSI network management model. The Management subtree is used to define objects in the Internet standard MIB. This consists of objects that are expected to be available on managed nodes running the Internet suite of protocols. The latest version of this MIB is referred to as MIB-II[4]. It contains 171 objects in a number of groups that are identified as System, Interfaces, Address Translation, IP, ICMP, TCP, UDP, EGP, transmission and SNMP. Each group is considered optional, but if any object in a group is implemented, then the whole group must be implemented. The Experimental subtree is used for conducting Internet experiments. The Private subtree allows any enterprise to register with the Internet community and build their own MIBs. MIBs developed by the SSC Lab fall into this category.

III. What is SNMP?

SNMP has become widely implemented in the network community as the accepted de facto standard network management protocol. Agents supporting SNMP are provided by many network device vendors. In addition, many workstation vendors provide an SNMP agent either as part of their standard operating system release or through third party vendors.

SNMP provides four operations: Get, GetNext, Set, and Trap[5]. It requires a connectionless transport service to be provided. TCP/IP implementations of SNMP use the User Datagram Protocol (UDP) as the transport mechanism. Community names provide for minimal authentication access to a managed nodes MIB. Different community names can be used for Get and Set to provide read-only access to some users and read-write access to others. Get and Set operate on a specified OID. GetNext returns the OID and value of the next object in the MIB. This allows a Network Management Station (NMS) to "walk" through a managed nodes MIB by repeatedly issuing GetNext calls to that node. Traps are the means that the managed node can report that an event has occured or some threshold has been passed.

Many commercially available NMSs are available that use SNMP. NMS provides a user interface to the SNMP objects and allows for data collection, MIB browsing, trap handling and maintaining graphical maps of the network. The NMS can draw conclusions on the health of the network and its associated devices based on the MIB data it collects. Additionally, SNMP applications can be purchased or written to manipulate the data in a number of ways. e.g. a meter to show the network or CPU utilization of a particular device.

IV. SNMP Agent for Real-Time Systems

In order to provide SNMP services for real-time systems, SNMP agent software was purchased in source form. This software is the SNMP Universal Agent[6]. C source code is provided to implement the SNMP agent, and sample MIB interfaces which simulate MIB-II and the experimental Uninterruptible Power Supply (UPS) MIB. Each MIB adheres to a defined Agent-MIB interface so that new MIB modules can be added. This code was ported to VxWorks[7], a real-time operating system, by SSC Lab engineers.

VxWorks provides many data structures that contain data relevant to the status of the network. A mapping of this data to the MIB-II objects was performed by SSC Laboratory personnel resulting in a near complete implementation of MIB-II under VxWorks.

V. SSCL Real-Time MIB

With the standard MIB-II objects in place, it is apparent that extensions are needed in order to manage real-time systems. Objects that are not part of any existing standard MIB are desired. For instance, from the central management station, it may be desireable to change a nodes configuration, determine the current software verison, modify the state of tasks in the system, determine the CPU load or even reboot the system.

A MIB was designed to address these unique requirements for management of real-time systems, referred to as the SSCL Real-Time MIB. There are four main groups in the MIB: Real-Time Operating System (RTOS), SNMP Daemon (SNMPD), CPU Idle (IDLE) and System.

The RTOS group contains objects relevant to each real-time operating system that the agent may be ported to. For example, a VxWorks sub-group could contain objects representing system memory usage, system tasks and boot parameters. The current implementation allows the VxWorks boot parameters to be interrogated and modified using SNMP. Due to the distribution of real-time systems at the SSC over many miles, this functionality could be invaluable.

The SNMPD group contains information relevant to the tasks used to support SNMP on the target. These objects include the version of the Agent core, version of the Operating System Port, and the task priority of the daemon task. The task priority is settable using SNMP.

The CPU idle group contains information regarding the utilization of the CPU. Objects provide the CPU percent idle at various time intervals as well as a user settable time interval. The values can be queried to monitor the system performance.

The system group is intended to monitor and control the real-time system as a whole. It contains objects that allow the user to start reboot sequences or abort reboot sequences to a target. When these values are set, SNMP Traps can be sent to a NMS to advise that the system is being rebooted. These traps contain the system being rebooted, how long until the reboot will occur and which system caused the reboot to occur.

VI. SSCL T1 MIB

The communications for the SSC controls system will consist largely of point-to-point links to satisfy the throughput and response time requirements[8]. These point-to-point links will be provided directly into the real-time system by means of a fractional T1 interface. This interface is being developed by the SSC to implement standard protocols such as HDLC, PPP, and TCP/IP. Some of these interfaces are already managed by standard MIBs defined by the Internet community. Where applicable, those MIBs will be used to manage the point-to-point links. In addition to that, a SSCL T1 MIB will be designed to implement direct driver level statistics about the T1 interfaces. This MIB will allow network managers to determine the operational status of the interfaces, and verify that the T1 communications are set up properly in accordance with ADM and SONET equipment used to transport the T1. For example, the T1 channels involved in a point-to-point link could be verified.

VII. Real-Time Performance issues

When real-time systems are operational, there are essentially three types of operations: Interrupt Service Routines (ISR), real-time tasks, and non real-time tasks. The goal of the real-time operating system is to schedule the real-time tasks in a deterministic nature. The non real-time tasks are tasks that are generally not mission critical (e.g. a user level shell), they may use the remaining CPU time, or if there is no remaining time, they are postponed until CPU resources are available.

Since the SNMPD task is not mission critical to the embedded system, but rather provides support information to the management station, it is considered part of the non real-time group of tasks. To address this issue, the SNMPD task has been designed to run at any priority level. The system designer assigns a priority when the SNMPD task is initialized, or also while it is running through the use of SNMP.

The local CPU utilization is only one concern for the impact of the SNMP task on the real-time system. The other consideration is the bandwidth requirement added to the network. In the SSC control system, each real-time system will have a fractional T1 interface. This means that a given system will have a dedicated bi-directional network bandwidth between 64 kbps and 1.554 Mbps. (i.e. the total bandwidth is available for transmitting and receiving data simultaneously). Although the exact traffic patterns for the SNMP data are not known at this time, estimates can be made based on experience using an existing NMS to manage ethernet networks. The NMS could query a node at some user configurable rate for the network utilization, CPU utilization and a few other parameters. This total of about 15 objects could be sufficient to determine if the system is functioning.

It is estimated that each object requires a packet of 128 bytes to be transmitted on the network with a response of the same size. Based on this packet size, SNMP would require 1920 bytes for a complete data acquisition cycle. If the polling period is set to one minute, then the theoretical bandwidth of a 64kbps channel is 480KB/minute, and the bandwidth of a T1 is 11.7MB/minute. The SNMP traffic is then 0.39 percent of the 64 kbps channel or only 0.016 percent of the full T1.

VIII. Future directions

The Real-Time MIB will be expanded to provide additional functionality in terms of Operating System configuration, utilization and task maintenance. Some of these areas will be threshold monitoring for various portions of the MIB including sending out SNMP traps when the CPU utilization passes certain thresholds. It is desireable to be able to set task priorities, interrogate memory, and possibly to use SNMP to implement some debug capabilities for the real-time system. Each of these areas are under investigation.

In the areas of network management protocols, the Internet Engineering task force is nearing completion of the next generation of SNMP, SNMP Version Two, commonly referred to as SNMPv2. SNMPv2 addresses manager-to-manager communications, bulk data transfer, and security enhancements. It is widely expected to be the replacement for SNMPv1. Because of this development, plans are already in place to move the SSC Lab real-time SNMP Agent to SNMPv2 compatibility.

It is not clear if or when the SSC control system network will migrate to an OSI-based network, but we believe that CMOT for real-time systems is feasible.

IX. Conclusions

SNMP seems quite well suited for managing real-time systems in the SSC Lab control system. It provides a common management protocol for traditional network devices as well as UNIX systems, real-time systems and possibly other control system components. SNMP's wide vendor acceptance provides a common ground for network management. This allows the SSC to manage its devices while still using commercial network management packages. With a sufficiently advanced NMS, proactive network management should be feasible. The background work put in by the Internet community will allow following the direction of the community as the standards migrate to newer versions of technology. This will certainly provide some interesting challenges in implementing the network management of the SSC controls system network.

References

[1] M. Rose, *The Simple Book*, Prentice-Hall, Inc., 1991, ISBN 0-13-812611-9.

[2] M. Rose and Keith McGloghrie. *Structure and Identification of Management Information for TCP/IP base Internets*, May 1990.

[3] M. Rose and Keith McGloghrie. *Concise MIB Definitions*, Request for Comments 1212, March 1991.

[4] M. Rose and Keith McGloghrie. *Management Informanation Base for Network Management of TCP/IP-based internets: MIB-II*, Request for Comments 1213, March 1991.

[5] J. Case, M. Fedor, M Schoffstall, J. Davin, *A Simple Network Management Protocol (SNMP)*. Request for Comments 1157, May, 1990.

[6] Paul Freeman Associates, 14 Pleasant Street, P.O. Box 2067, Westford, MA 01886.

[7] Wind River Systems, Inc., 1010 Atlantic Ave., Alameda, CA 94501.

[8] S. Hunt, C. Kalbfleisch, K. Low, D. Mathieson, *"BUBBANET" - A High Performance Network for the SSC Accelerator Control System*, these proceedings.