PRIME Research Final Report

Nathan Fry

Dylan Jones

Matt Matteson

June 21st, 2007

Table of Contents

Executive Summary

The goal of this project is to develop a library for virtual devices to utilize Simple Network Management Protocol (SNMP.) These devices exist within the Scalable Simulation Framework Network (SSFNet) and will monitor and store information related to their configuration and activity. We will further develop a management information base (MIB) implementation which will provide network management data and an interface for updating any mutable entries. Our goals were: setting up methods to handle different data types that will be handled by SNMP, and use commercially available products, such as the Multi Router Traffic Grapher (MRTG), to demonstrate our implementation with graphical representations of critical SSFNet activity

We have created a well structured implementation of the SNMP protocol and architecture. We were able to set up the structures to allow proper handling of a spectrum of different data types, which provided the basis for all future work on this project. We also implemented a well-formatted MIB-2.interface that supplies the underlying structure for devices to place and update data relevant to their operating status. Testing our implementation with standard SNMP software packages was attempted, but was unsuccessful as the emulation part of the simulator was not operable. Testing was performed via a simple two host simulation with one host performing queries on our simulated agent. This addition of SNMP provides a bridge to many existing network utilities.

Abstract

The goal of this project is to develop a library for virtual devices to utilize Simple Network Management Protocol (SNMP.) These devices exist within the Scalable Simulation Framework Network (SSFNet) and will monitor and store information related to their configuration and activity. We will further develop a management information base (MIB) implementation which will provide network management data and an interface for updating any mutable entries. Our goals were: setting up methods to handle different data types that will be handled by SNMP, and use commercially available products, such as the Multi Router Traffic Grapher (MRTG), to demonstrate our implementation with graphical representations of critical SSFNet activity.

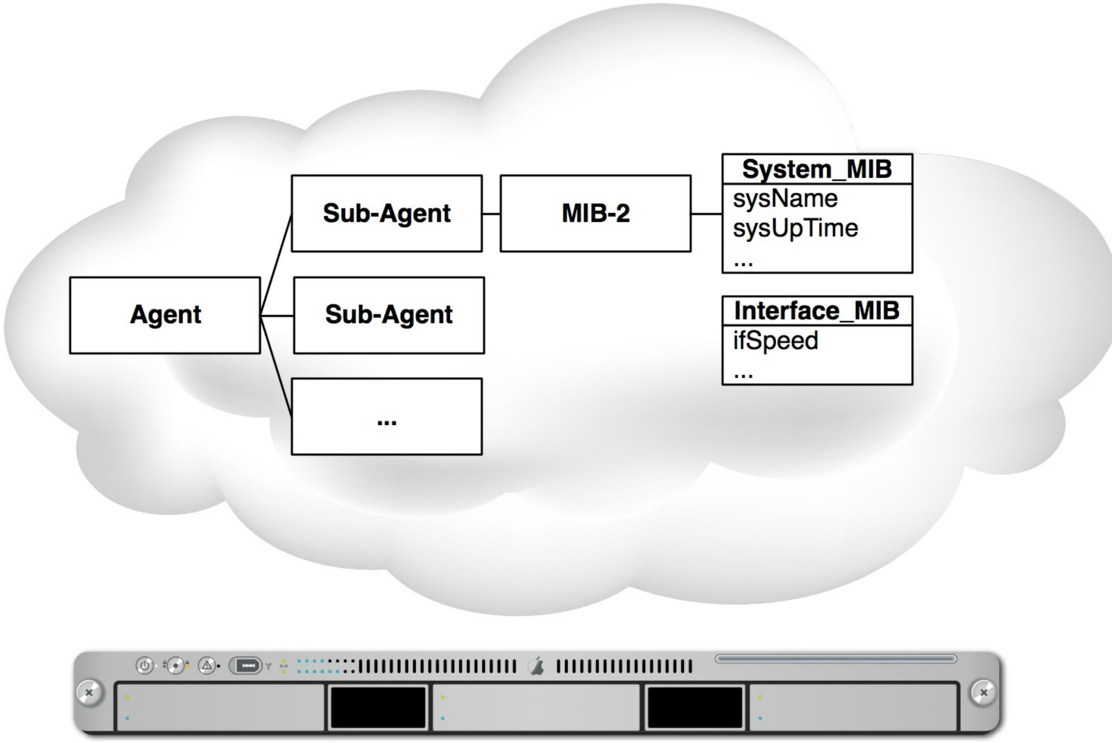
Introduction to the Project

The goal for our project was to develop the functionality of SSFNet in order to aid in the monitoring of the simulated network by providing SNMP interoperability. Programmed mostly in C++, the latest source code was provided to us at the beginning of field session. Configuration of SSFNet is done by defining a network in files using Data Manipulation Language (DML.) The implementation of SNMP will provide a simple and standardized way to monitor and update devices in the network.

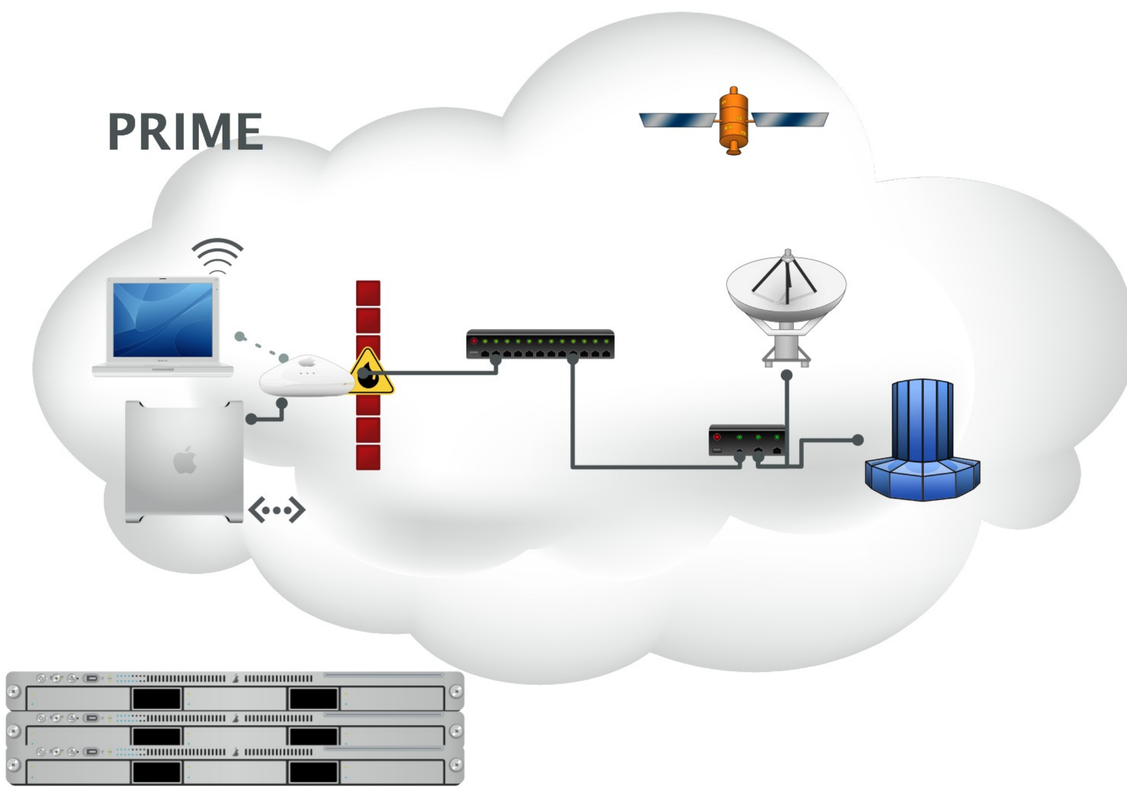
Requirements and Specifications

SNMP defines an architecture and protocol that must be implemented to ensure interoperability. Our code matches these requirements by implementing a master agent that communicates with a management station to receive requests. The master agent then queries subagents to obtain the information the management station requires. The subagent provides this functionality by maintaining a Management Information Base (MIB,) which stores managed objects of characteristics of the device.

The SNMP implementation must match the SSFNet requirements as to maintain distributive ability. The project must provide an interface for updating mutable entries within the MIB.



(Fig. 1 - SNMP architecture on a device.)



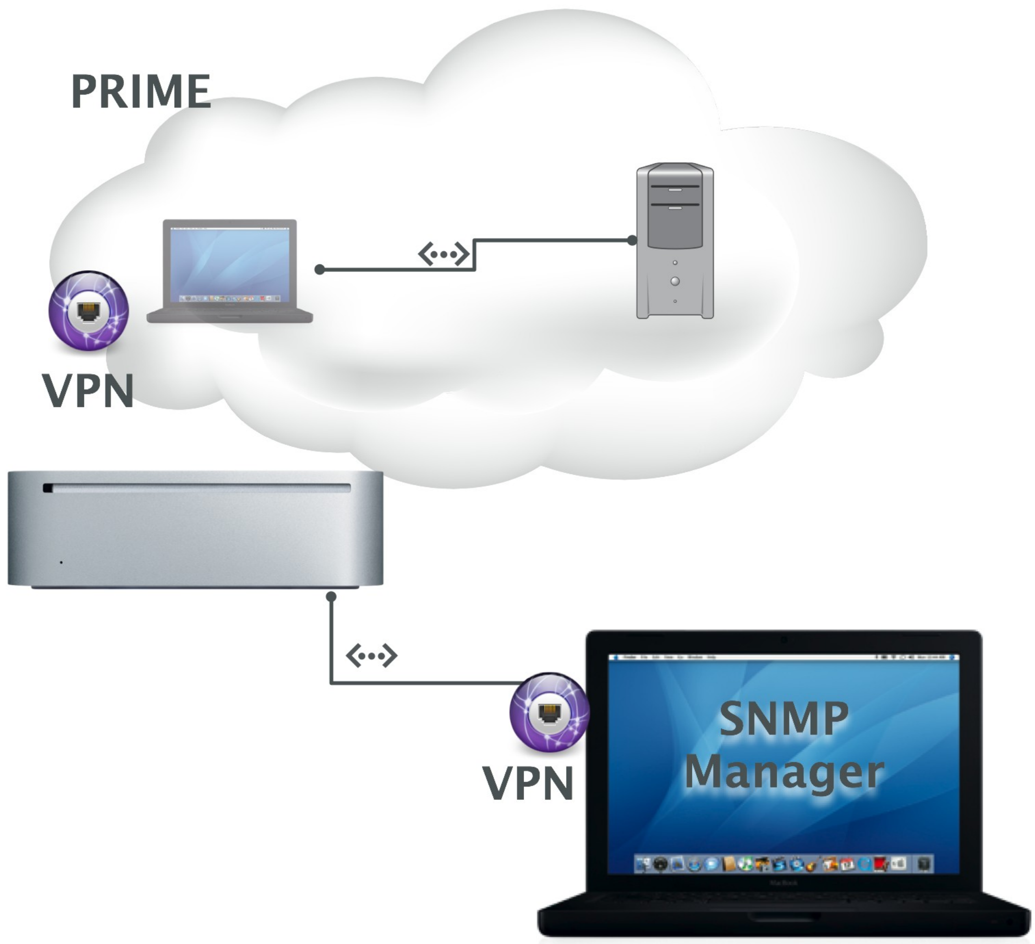
(Fig. 2 - PRIME simulating a large network on a cluster.)

Design, Implementation Details, and Results

In order to allow the SNMP agent to handle the required types of data, modifications were made to the types file to ensure it contains all the ASN.1 classes of implemented data types. By doing this, we have given our agent the ability to handle a multitude of different objects in the MIB. All commonly used data types (string, boolean, null) have been implemented in our project, and provide an easy process of implementing any future types.

We have also constructed a MIB to maintain objects of the system. Values such as device name are obtained when the MIB is loaded, and the MIB defines the value by reading from the configuration file. Other values are maintained dynamically by talking directly with other classes. An example of this would be the ifSpeed (interface speed,) when the ifSpeed object is requested from the MIB, the details are obtained by calling a function of the interface class.

To support the goal of SNMP interoperability, testing of our implementation of the protocol was held to the industry standard. There is no way to run a binary application within the simulator, so an emulation bridge must be set up. This hybrid network setup allows real computers and applications to share a network with virtual devices. This is done using an OpenVPN gateway and two computers. The creation of this technology is not part of our scope, and is very experimental at its current state. Learning and setting up this hybrid structure with industry standard SNMP managers is a way to test our implementation, while also using our structures to examine the inside of the simulated network.



(Fig. 3 - Our physical setup, bridging a simulated network in PRIME to a real device running standard SNMP software.)

The hybrid network testbed became a burden on our project. The current state of the emulation did not provide any documentation and direct communication was the only route for even attempting the configuration. We have not been successful with the sample hybrid network setup, and thus unable to test our implementation with standard applications.

Conclusions

The result of our project was implementing the SNMP architecture to allow interoperability to manage and query devices in SSFNet. The design of our implementation supports the common functionality of SNMP while providing an expandable base of code. New data types and objects can be defined using our set style to easily match any unpredicted uses of the project. As the hybrid functionality develops, our SNMP functionality will allow using proven sampling and configuration tools to further the abilities of the simulation.

Glossary

SNMP - Simple Network Management Protocol

SSF - Scalable Simulation Framework

MIB - Management Information Base

VPN - Virtual Private Network

IP - Internet Protocol

DML - Data Manipulation Language

ASN.1 - Abstract Syntax Notation #1

PRIME - Parallel Real-time Immersive Network Modeling Environment

References

[1] http://prime.mines.edu/

[2] http://asn1.elibel.tm.fr/

[3] http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm

[4] http://openvpn.net/