APPLICATION OF THE SHAMAN MANAGEMENT SYSTEM TO CONFIGURATION MANAGEMENT OF TACTICAL INTERNETS

Adarshpal S. Sethi **Dong Zhu** **Vasil Hnatyshin** **Prashanth Kokati**

Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716
{*sethi, dzhu, vasil, kokati*}*@cis.udel.edu*

Deh-Phone Hsing **Latha Kant**

Telcordia Technologies, Inc.
445 South Street, Morristown, NJ 07960
{*dhsing, lkant*}*@telcordia.com*

ABSTRACT

SHAMAN (Spreadsheet-based Hierarchical Architecture for MANagement) is a novel management framework developed at the University of Delaware; it extends the traditional flat SNMP management model to a hierarchical architecture. Previous papers at the ARL/ATIRP Annual Conferences have described the framework, its prototype implementation, and a location management application for tactical battlefields. The SHAMAN system is being used within a joint Task demo for adaptive configuration management by reconfiguring network connectivity when node positions have changed. This paper presents an overview of SHAMAN and the role played by it in the joint demo for configuration management.

Keywords: Network Management, Hierarchical Management, SNMP, Tactical Internet, Configuration Management.

I. INTRODUCTION

One of the significant achievements of the ATIRP Consortium in Technical Factor 2 (Tactical/Strategic Interoperability) has been the design and development of an integrated framework for hierarchical management called SHAMAN (**S**preadsheet-based **H**ierarchical **A**rchitecture for **MAN**agement). This management system developed at the Network Management Laboratory of the University of Delaware incorporates management by delegation concepts into the Internet management framework to facilitate the management of distributed systems [1], [2], [3], [4]. This architecture allows a manager to delegate routine management tasks to an intermediate manager and facilitates user configurability of management information and control in a value-added manner.

Among the many challenges faced in the management of tactical battlefield networks is the issue of reconfiguring the network connectivity in the face of node mobility, fluctuation in availability of communication channels due to varying topology, and loss of nodes and communications because of jamming or enemy activity. It is thus important to design a management system for auto-configuration that can adapt to and respond to dynamic network conditions. The hierarchical architecture of SHAMAN lends itself to application in this environment.

This paper describes the application of the SHAMAN system to implement a topology reconfiguration algorithm developed at Telcordia [5]. This application is being developed as part of a joint task demo between the collaborating partners of Task 2b on "Adaptive Network Configuration and QoS Management" with the objective of showcasing the technology developed in this task over the lifetime of the ATIRP Consortium. Section 2 of the paper first provides a brief overview of the SHAMAN system. Section 3 describes the plan for the joint demo with an

Prepared through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program Cooperative Agreement DAAL01-96-2-0002.

emphasis on the interactions between the SHAMAN system and the other subsystems in the demo. Section 4 outlines how the reconfiguration algorithm is being implemented in SHAMAN, and Section 5 presents the conclusions.

II. Overview of SHAMAN

A hierarchical management strategy is an effective means of managing the large and complex internetworks that are in use today [6]. The need for hierarchical management is particularly acute in tactical battlefield networks which are expected to have tens of thousands of nodes, and where scalability is an important concern. Unfortunately, the most popular management framework in use today, the SNMP framework [7], [8], only supports the flat management model. The framework provides no means for managers to delegate tasks to intermediate managers or for peer-to-peer communication between intermediate managers during the execution of these tasks. SHAMAN provides this much-needed capability to the Internet management framework.

SHAMAN allows a manager to delegate tasks to an intermediate manager (IM) by downloading scripts expressing these tasks into a spreadsheet-like structure of the IM called the Spreadsheet MIB [9], [10], [11]. This MIB is divided into a two-dimensional structure of cells called a spreadsheet, with each cell having a control part that stores the script and a data part that contains the result of executing the script. One IM can support multiple spreadsheets.

The spreadsheet MIB implements the spreadsheets using SNMP tables. All operations on the spreadsheet including a manager's downloading of scripts into the control parts and accessing the results in the data part are carried out via the Get and Set operations of the SNMP protocol. User operations on cells map to operations on tables that are part of this MIB. When the IM receives an SNMP request from the manager that translates to an operation on a cell, the IM performs the necessary operations on the spreadsheet MIB to implement the cell abstraction. Once the request has been carried out, the IM returns a response to the manager that requested the cell operation.

The scripts in SHAMAN are written in a special

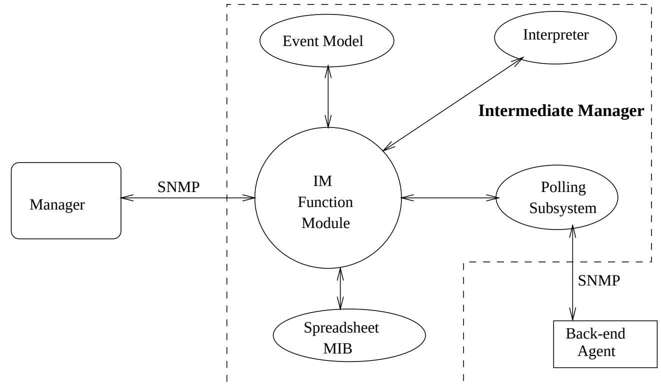

Fig. 1. Prototype Implementation of SHAMAN

language called the Spreadsheet Scripting Language (SSL). This interpreted language contains features that facilitate the development of procedural scripts as well as event specifications. The language includes

- procedural language related features including operators, variable support, and control flow constructs
- network management specific features including polling specification and management variable access
- paradigm specific features including cell access, retrieval, modification, and multiple value access
- event model related features including event and event dependency specification

A prototype implementation of SHAMAN has been developed at the University of Delaware; the first version dubbed Version 1.0 contained most of the basic features of the SSL. Version 2.0 of SHAMAN has since been completed and contains some enhanced features that aid application development and also permit some peer-to-peer interactions. Both versions of SHAMAN are available on the WWW at the URL

http://www.cis.udel.edu/~shaman

Figure 1 shows the software architecture of the IM and the interdependencies of the various modules that constitute the IM. Among these modules, the MIB Module, the Interpreter Module, and the Cell Module together implement three of the logical components of the IM. The Polling Subsystem implements the polling of the agents. The other modules perform support functions like timer services and providing a communication interface for polling the agents.

We have also developed a demo application of

SHAMAN that illustrates its use in a hypothetical scenario of location management for mobile nodes in a battlefield network [4], [10]. This application was presented and demonstrated at previous ATIRP Annual Conferences. It uses two Intermediate Managers, each running an instance of SHAMAN, to poll and manage nodes as they move on the battlefield. When a node moves out of the domain of an IM, it initiates a handoff to the other IM via a message to the top-level manager. The tasks executed at both IMs are coded as SSL scripts and loaded into their Spreadsheet MIBs by the top-level manager.

III. Joint Demo for Adaptive Reconfiguration Management

The joint demo among all the task members is being planned to highlight the significant achievements of this task over the past five years in the area of tactical network management and interoperability. These achievements include the SHAMAN system for hierarchical management of tactical networks, the adaptive reconfiguration algorithm, and a multi-level distributed "manager/agency" paradigm to support fault-tolerant and scalable network management and control.

The joint demo adopts a CORBA-based network management system for the overall top-level management of tactical networks. The CORBA architecture supports distributed object computing in a distributed heterogeneous environment. Interface Definition Language (IDL) is used to define the services offered by an object. CORBA provides a flexible platform for developing management applications for the overall control of the networks being managed and provides the vehicle for interaction and communication between the top-level manager, remote collection stations, intelligent agents, and mid-level managers. It should be noted that the CORBA platform is used as the basic software platform for the managers, whereas a network management system using SNMP is envisioned to be used between the managers and the managed network elements in the IP-based tactical internet. This latter management system would include the deployment of multiple SHAMAN entities at various levels to provide for a hierarchical management structure.

Some of the key objectives of the joint demo are:

- Demonstrate the scalability of the proposed network management architecture.
- Demonstrate the support of heterogeneous network management technologies.
- Demonstrate the adaptability to changing network requirements and conditions.
- Demonstrate network management automation capabilities.

The joint demo assumes that the degree of network node mobility anticipated in tactical networks is not uniform, and that the network nodes at the division level and above are relatively stationary. However, the mobility of the network nodes increases and becomes less predictable as we move closer to the war fighters. It is thus assumed that there is a tactical backbone consisting of ATM switches which is only subject to pre-planned reconfiguration due to mission re-assignment. The Brigade level and below consists of a set of IP-router based tactical Internets interconnected by mobile switching centers with ATM switches which also provide the connectivity to the backbone. A manager-agency paradigm is assumed in which the top-level manager is given overall network management responsibility of the entire division. This manager can delegate tasks to corresponding Intermediate Managers (IMs) in the Tactical Internets (TIs).

The top-level manager is developed as a CORBA object which will communicate with other CORBA objects to access an LDAP directory service system (DSS). This manager runs the configuration algorithm to determine the connectivity among the tactical backbone and the TIs. It then delegates responsibility to the SHAMAN-based IMs in the TIs for running the configuration algorithm to determine the connectivity among the nodes within its delegated management domain.

IV. Configuration Management with SHAMAN

Version 1.0 of SHAMAN implemented only the basic features of the Spreadsheet Scripting Language (SSL) in order to demonstrate proof-of-concept. These features were sufficient to implement the Location Management demo, although the demo could be streamlined and made more efficient if additional features were present. However, the additional features are required if SHAMAN is to be applied to more complex situations such as configuration management. In the past year, we have completed SHAMAN Version 2.0 which includes the features briefly described below:

- Revision of cell structure to allow table storage.
- Dynamic specification of cell names.
- Dynamic specification of OIDs and hostnames.
- Foreachrow and Foreachcol statements.
- Synchronous operation of SNMP Get and Set statements.

The availability of the above features now makes it possible for SHAMAN to be used in both manager and agent roles dynamically. Thus, peer-to-peer operations are now feasible so that SHAMAN need no longer be operated only in a strictly hierarchical manner. The dynamic specification of MIB variables and of cells is very valuable for configuration management in dealing with a multitude of nodes, agents, and variables.

The topology reconfiguration algorithm designed by Telcordia [12], [5] applies a number of heuristics to generate connectivity within a TI. The heuristics are based on using the well-known Dijkstra's shortest-path algorithm to compute the shortest path tree from a given node to all other nodes within the network. This shortest path algorithm is modified to accommodate the constraints that are imposed on the configuration by the application. Three constraints that are used are:

- Every node in the network must be able to communicate with every other node via at least one path that is less than H hops long.
- Every communicating node pair has more than one possible routes through the underlying network (to provide redundancy in case there are failures).
- There is a a maximum limit on the number of other nodes with which any given node has direct connectivity.

To implement this algorithm in SHAMAN, we have planned to use a structure of cells in the spreadsheet which will make it easy to modularize the different logical components of the algorithm. The first column of cells are used for external interaction from the SHAMAN entity, i.e., for communication with the top-level manager. This column includes a cell used as the trigger for topology generation. The manager will do an SNMPGet operation on this cell to command the SHAMAN IM to begin topology reconfiguration when it is necessary to do so. Another cell is used to store the generated topology; this topology is then set into an agent MIB that is used to drive the input to the directory service system from where it is accessible to the manager.

A second column of cells stores global parameters and data. These include the number of nodes in the network, the node addresses, the maximum limit on nodal connectivity, and the limit on the maximum length of the shortest path between two nodes. The third column contains local structures to be used by the various components of the algorithm. These are the node positions (which are polled from the agents in the nodes, or may be obtained from the DSS), the path cost matrix, the direct link costs, and the predecessor of each node in the shortest path tree.

The remaining columns and cells in SHAMAN's spreadsheet are used for the code corresponding to the algorithm's components. The first of these is the Chain generation function, which initially generates chains between the nodes that are as long as possible without violating the maximum length constraints. The second is Dijkstra's algorithm which finds the shortest paths from each node to every other node. The third component is a function to create shorter paths whenever the algorithm has paths that are unacceptably long. This is done by adding direct links between the nodes so that path lengths may be reduced. The final component is a function to perform checks on the nodal degree constraint.

Implementation of this algorithm in SHAMAN is currently underway and we hope to be able to give a demonsration of the SHAMAN part of the system at the upcoming Annual Conference. The integration of this with the rest of the system being developed by our other task partners will be done shortly afterwards, and the complete joint demo should be available for demonstration before the end of the project.

V. Conclusions

In conclusion, one of the significant achievements of the ATIRP Consortium in the area of network management has been the development of the SHAMAN system. SHAMAN is an architecture and framework for hierarchical management of networks that can be applied to implement various management applications in the Army's tactical battlefield networks. We have previously demonstrated its utility for location management of mobile nodes in a battlefield environment. This paper has described the development of a configuration management application using SHAMAN that can be used to reconfigure network topology when node positions have changed. A prototype implementation of SHAMAN is available and will soon be ported to the ARL Testbed so it can be used for other management tasks and applications within the Testbed.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

REFERENCES

[1] P. Kalyanasundaram, A.S. Sethi, and C. Sherwin. Design of A Spreadsheet Paradigm for Network Management. In *Proceedings of the 7th IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, L'Aquila, Italy, October 1996.

[2] P. Kalyanasundaram, A.S. Sethi, C. Sherwin, and D. Zhu. A Spreadsheet-based Scripting Environment for SNMP. In A. Lazar, R. Saracco, and R. Stadler, editors, *Integrated Network Management V*, pages 752–765. Chapman and Hall, London, 1997.

[3] A.S. Sethi, P. Kalyanasundaram, C. Sherwin, and D. Zhu. A Hierarchical Management Framework for Battlefield Network Management. In *Proceedings of MILCOM '97, IEEE Military Communications Conference*, Monterey, CA, November 1997.

[4] A.S. Sethi, P. Kalyanasundaram, C. Sherwin, D. Zhu, and S. Pakala. Battlefield location management applications of SHAMAN. In *Proceedings of MILCOM '98, IEEE Military Communications Con ference*, Boston, MA, November 1998.

[5] L. Kant, C.-H. Zhu, D.-P. Hsing, and M. Lee. An adaptive configuration management architecture design for the army's networks. In *Proceedings of the Fourth ARL/ATIRP Annual Conference*, pages 223–227, College Park, MD, March 2000.

[6] A.S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors. *Integrated Network Management IV*. Chapman and Hall, London, 1995.

[7] J. D. Case, M. S. Fedor, M. L. Schoffstall, and C. Davin. *Simple Network Management Protocol (RFC 1157)*, May 1990.

[8] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2) (RFC 1905)*, January 1996.

[9] A.S. Sethi, P. Kalyanasundaram, C. Sherwin, and D. Zhu. A Spreadsheet-Based SNMP Scripting Environment for Battlefield Network Management. In *Proceedings of the First ARL/ATIRP Annual Conference*, pages 251–256, College Park, MD, January 1997.

[10] A.S. Sethi, P. Kalyanasundaram, C. Sherwin, and D. Zhu. Battlefield applications of hierarchical management with SHAMAN. In *Proceedings of the Second ARL/ATIRP Annual Conference*, pages 235–240, College Park, MD, February 1998.

[11] A.S. Sethi and D. Zhu. Debugging scripting applications fo battlefield network management. In *Proceedings of the Third ARL/ATIRP Annual Conference*, pages 177–182, College Park, MD, February 1999.

[12] A.S. Sethi, D. Zhu, Y. Zong, D.-P. Hsing, and L. Kant. Application of the shaman management system to battlefield network configuration management. In *Proceedings of the Fourth ARL/ATIRP Annual Conference*, pages 187–191, College Park, MD, March 2000.