Application of Mobile Components for Robust Network Management

Madhuranath Konety[1], Raghu Dev[2,] I.R. Chen[3], S. Gupta[3]

Abstract-- **The dependence on networking for communication and information exchange in real-time conditions is extensive and very important. Ensuring the operability and availability of these networks is a challenge for most network managers. We address the growing demands from network management systems as the complexity of network services increase with the arrival of new technologies. This work illustrates a novel scheme where management systems can be built from a set of independent components to address specific functional areas. These functional components can be distributed and also migrated to address management needs in adverse conditions.**

Keywords--Network Management, Active Networks, High Availability, Mobile Components, Distribution

I. Introduction

The availability of accurate and current information in real-time environments such as military operations is extremely important and is heavily dependent on the communication networks that provide access between diverse and distributed sources of information. These networks are evolving to meet growing demands of services and availability with the addition of new technologies such as High Speed Networks, Wireless Communication, Active Networking. The need to manage these highly complex systems is vital.

The manager-agent technology used by most present day network management approaches such as the Simple Network Management Protocol (SNMP)[4,5,6], the OSI Common Management Information Protocol (CMIP)[7] in data networking and the Telephony Management Network framework (TMN)[12] in the telephony area, address traditional management requirements. The emphasis is on the simple distribution of functionality between managers and agents, where agents interact with the managed resource to provide the interface between the resource and the management system. A manager interacts with several agents to provide a centralized single point of interface to the network.

Management of these evolving networking technologies has to address several issues such as robustness, adaptive and dynamic management [17]. Robustness can be addressed as a combination of high availability, fault tolerance and disaster recovery. The areas of fault tolerance and disaster recovery are addressed at the hardware and system levels. We address the issue of high availability with application of a distributed paradigm. Distribution in current network management techniques is addressed by enhancing the capabilities of the agents or by introducing new managers at intermediate points [3]. The use of multiple managers in either a peer to peer or a hierarchical architecture is not scalable beyond a threshold. Adaptive management is an area that benefits greatly from the advent of new technologies, network nodes have become powerful enough to not only participate as communication devices but also perform computations as with Active Networks [13]. This leads to a system that can be dynamically reconfigured and adapted to changing conditions. The Active Networks technology and program mobility leads to the mobility of the management, where managed areas can be enhanced with the required functionality to dynamic management. In traditional manager-agent frameworks, functionality can not be easily migrated or extended, as they are concentrated at specific points.

In this work we address the specific issues of management needs of high availability, adaptive and dynamic management. We propose a model where we identify network management functionality as separate from the supporting services. These can be decomposed into functions that are represented as components. This leads to a simple scheme for functionality distribution independent of the underlying technologies. In the next section we highlight some of the popular efforts for network management and the effects on robustness of these approaches. In section 3 we describe our mobile component model and the distribution goals. Section 4

[1] madhu.konety@stdc.com Sr.Software Engineer System/Technology Development Corporation, Herndon VA
[2] rdev@lucent.com System Engineer APX Technologies Columbus OH
[3] [irchen,sgupta]@vt.edu Dept. Of Computer Science, Virginia Polytechnic and State University, Falls Church, VA

summarizes the work and provides insight into some possible future research areas.

II. Robustness in Prevalent Network Management System Architectures

A typical network management model consists of several element managers that manage a well-defined subset of the network [3,8]. Element management systems in turn describe a manager/agent model that consists of various components including a user interface, a map of the resources being managed, a protocol for querying the state of the managed resources and a database management system that stores historical and real-time information. The manager/agent model gives rise to three different architectures, Centralized, Hierarchical, and Distributed. These architectures fail to be robust and hence cannot provide 24x7 service military type systems. The centralized and the hierarchical architectures suffer from single point failures. The few available solutions for these architectures are usually cost prohibitive. Example: Most suited and usually employed is N+1 cold/warm/hot sparing. Having an extra resource either with partial or no load ready to assume the identity of the failed machine. The switchover can be automatic/manual, in both cases the time to restore the service is equal to the sum of the time to start the extra processor as the failed one plus the time to restart the service software.

In the distributed architecture databases are fully distributed to network nodes. This scheme is a combination of the two previous architectures. While it offers the advantages of fault tolerance and reduced network traffic, it requires the use of an efficient database replication and allocation scheme in order to minimize the system control function overhead. Only this ensures high availability. This scheme introduces data synchronization problems. Processes that perform these functions are usually very complex and have to be performed by the NMS over and beyond its normal functions.

III. Enabling Technologies for a Robust NMS

A novel approach to network management has been proposed in [9,10]. This approach uses a combination of management delegation and extensible agent technology. Delegation involves the transfer of management functionality to an agent that is extensible, i.e., being able to dynamically incorporate new functions/services if needed. The most striking advantage of this approach is its ability to deploy management solutions to where the data reside. This is in direct contrast to traditional models where data must be extracted first and then a decision is made. One disadvantage of this approach is that the environment must support the extensible agent infrastructure. Secondly, there is a need to correlate information across agents. This need necessitates the use of a central manager to access information across agents.

A comprehensive survey on approaches to network management based on active networking appears in [2]. Active networking is a new paradigm where network nodes can process not only data packets for normal data communication, but also active packets for executing embedded programs.

In the ``smart packet" approach [11], a program is fully contained by a sequence of active packets. An active node receives these packets, executes the embedded program and a returns a value. In this paradigm, a manager can send programs to nodes close to a problem area to perform localized management functions. A program can also be extended to include schemes to enable problem fixes, thus leading to an automated and self-enabled management system. It provides a practical scheme by the use of small sized programs (under 1 Kbytes). A drawback with this approach, however, is that the state information on active nodes is not saved between executions. Thus, persistent or recurrent problems may not be solved efficiently. It also incurs some overhead in assimilating information in every execution. Moreover, it does not address the issue of managing distinct devices and networks. In [1], applications of active networking to network management are described. The idea is to distribute and migrate program segments using active packets. However, no specific management problem is given as an illustration and a scheme to distribute the management process is not provided.

The rapid growth of Middleware applications such as CORBA [15] and COM/DCOM [16] provide extensive frameworks for distributed object and components based systems. The need for platform independent communication schemes is essential for network management systems to manage infrastructures that are truly heterogeneous in technologies, approaches and vendors. This heterogeneity also requires technologies that provide device independence for the various process components as in Java [14].

Although, individually these techniques have disadvantages as a combination they prove to be very useful. We take full benefit from the code mobility and dynamic deployment of services to produce a robust NMS. This would have mobile components so as to provide uninterrupted and highly available services as desired in a military operation.

IV. Mobile Component Model

A Network Management System provides one or more core functions such as fault, configuration, accounting, performance and security (FCAPS) functions [3,12]. To realize such a system, a set of base modules not part of the core functions are required. These modules provide certain basic operating system, networking, and data processing functionality to support the core management functions. A typical set of support modules functionality include *Communication*, *Data Handling (for Data Storage and Retrieval)*, *Security*, and *User Interface*. These modules are implementation specific and could vary from one system to another. Functionally, the structure of the NMS can be divided into two layers. The low layer includes essential base modules while the high layer are network management functions (NMFs) implemented using services provided by the lower layer modules. By separating the two layers and specifying interfaces between them, the NMS functionality can be distributed independent of the modules. Figure1 illustrates the relationship between the lower layer base modules and the high-level core NMS functions.

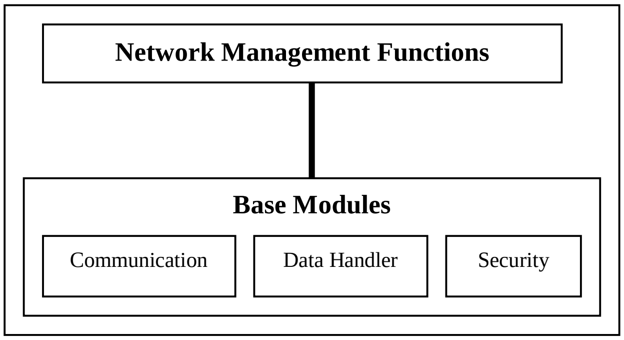

Figure 1: Management Functions and Base Modules

A. FUNCTIONS and COMPONENTS

The network management functional areas can be recursively decomposed into a set of base functions. The goal of the decomposition process is to generate a set of independent functions, where a process component can be instantiated to provide the functionality independent of all other functions. The required functional areas are thus a collation of these base functions. The top-level network management functional areas such as Fault management, Performance management and Configuration management are each individually decomposed to generate the respective base function sets. The depth of the decomposition process is based on the complexity that is sustainable by a single base function process and is at this point an intuitive process. The decomposition process produces functions that when distributed across nodes can produce results similar to a centralized system. However the decomposition and distribution ensure a higher level of reliability and availability.

An illustration of the decomposition process would be to provide only Health and Alarm surveillance under Fault Management, the required atomic functions could be event detection, reporting and logging as seen below.

$$F_{nm} = \{F_f, F_c, F_a, F_p, F_s\}$$
F_f = Fault management functions
F_c = Configuration management functions
F_a = Accounting management functions
F_p = Performance management functions
F_s = Security management functions

$$F_f = \{f_{as}, f_{fi}, f_{fc} \ldots \}$$
f_{as} = Alarm Surveillance function set
f_{fi} = Fault Isolation function set
f_{fc} = Fault Correction function set

$$f_{as} = \{ f_{as1}, f_{as2}, f_{as3}, f_{as4}\}$$
f_{as1} = Alarm Detection
f_{as2} = Logging
f_{as3} = Alarm Reporting
f_{as4} = Alarm Correlation

A key aspect of the functional decomposition process is that it could results in function sets that are not mutually exclusive. An example of this is most of the management areas would produce Logging and reporting as a base function. Such functions which address more than one functional areas are identified and can be implemented only once.

The base functions produced by the decomposition process are implemented as components. An individual component would address a single independent function. A component can be defined as an independent process entity that has a definite set of functionality defined and has adequate operational data. Thus a single base function can now be made available at distinct and separate points in the network as components. This distribution of the functional areas using components is a move towards improving the availability of a function. The non-availability of a function at some point in the network does not lead to the non-availability of the function for management itself. The functionality required can still be accessed at the other distributed components. The other aspect of the model that contributes to the robustness of the system is introduction of mobility to the components. The components are mobile process based on the following definitions:

- Dynamic Instantiation of a component on a node by transfer of the component byte codes to the node at run time
- Run time transfer of a component between two nodes including existing operational data

From the first definition, a component is now an active process entity. Components can be instantiated on various nodes at run time evenly autonomously. This is a departure from traditional schemes where management processes have fixed hardware locations to execute on. This definition provides the flexibility to provide management dynamically and to address needs dependent on network conditions. The second definition similar to those of Mobile agents []. A component can move from one node to another and resume it execution. The key to this definition is the transfer and availability of the components operational data.

An essential part of the component architecture is a scheme for controlling the distributed components. The distributed controller, a set of process replicated among the

0-7803-5538-5/99/$10.00 (c) 1999 IEEE

network nodes with each controller is responsible for the administration of the local components only. The controllers are responsible for:

1. maintaining the distribution of the components
2. providing communication between the components
3. handling the re-adjustment and reconfiguration of the components

An important task performed by such distributed control is to distribute sufficiently replicated components in the network. The components are initially allocated to the nodes based on an *initial placement* algorithm. The goal of the algorithm is providing a mapping of the various components on to the nodes with the available static information. The distribution of the components has to lead to a stable condition that will be investigated in a future work. The key factor to ensure is that adequate availability of the components with minimal replication. The key factor determining the Initial distribution includes static response time measures and also node factors including:

- Node resource availability: This factor includes the hardware resources of the network node itself.
- Node network resource availability : This factor includes the bandwidth and network connection resources

The controllers exchange run-time load information with others and based on this information dynamically reallocates components. A runtime readjustment algorithm with a goal to meet response time requirements with respect to real-time events controls this readjustment decision.

The distributed controller maintains locally a list of components allocated to the particular node. The controllers can communicate with controller on other nodes to exchange node resource information. Also the controller maintains a list of components that can provide a service not present on a particular node. This combination of information is then used to generate new allocations as part of a dynamic readjustment strategy to adjust to changing network conditions.

V. Conclusion

The distributed mobile component model is a step towards providing a robust management scheme for networks in real time environments. The distribution of the components is an approach to replicate and increase the availability of the management and control in these environments. The need to dynamically relocate components to adapt to changing network conditions is essential in providing the robustness required. The next phase of this work is to build an adaptive reconfiguration algorithm that will lead to the dynamic readjustment of the components to accommodate changes. The functional decomposition process is important to ensure the independence of the components and is an area for future work.

VI. References

[1] D. Raz and Y. Shavitt, ``An active network approach to efficient network management," http://cm.bell-labs.com/cm/ss/who/shavitt/pub/act.ps.

[2] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, G.J. Minden, ``A survey of active network research," *IEEE Communications Magazine*, Vol. 35, No. 1, Jan. 1997, pp. 80-86, http://www.tns.lcs.mit.edu/publications/ieeecomms97.html.

[3] A. Leinwand and K.F. Conroy, *Network Management: A Practical Perspective*, 2nd Edition, Addison Wesley, 1996.

[4] M. Rose and K. McCloghrie, *Structure and Identification of Management Information for TCP/IP based Internets (RFC 1155)*, May 1990.

[5] K. McCloghrie and M. Rose, *Management Information Base for Network Management of TCP/IP based Internets: MIB-II (RFC 1213)*, March 1991.

[6] J. Case, M Fedor, M. Schoffstall and J. Davin, *Simple Network Management Protocol (RFC1157)*, May 1990.

[7] U. Warrier, L. Besaw, L. LaBarre and B. Handspicker, *The Common Information Services and Protocols for the Internet (CMOT and CMIP) (RFC 1189)*, Oct. 1990.

[8] S. Feit, *SNMP: A Guide To Network Management*, McGraw Hill, 1995.

[9] Y. Yemini, G. Goldszmidt and S. Yemini, ``Network management by delegation," *2nd International Symposium on Integrated Network Management*, Washington, DC, April 1991.

[10] G. Goldszmidt and Y. Yemini, ``Distributed management by delegation," *15th International Conference on Distributed Computing Systems*, June 1995.

[11] B. Schwartz, W. Zhou, A.W. Jackson, W.T. Strayer, D. Roackwell, and C. Partridge, Smart Packets for Active Networks.

[12] *ITU-T Telecommunication Standardization Sector of ITU*, Series M: TMN Management Functions, M.3400.

[13] D. L. Tennenhouse and D. J. Wetherall,"Towards an Active Network Architecture", "Computer Communication Review",Vol. 26 No. 2, April 1996, "URL http://www.tns.lcs.mit.edu/publications/ccr96.html"

[14] Sunsoft, "The Java Language Specification", Sun Microsystem Inc., 2550 Gracia Ave, Mountain View, CA 94043, USA, 1995, http://www.javasoft.com/products/jdk

[15] Object Management Group, "Common Object Request Broker Archcture", OMG Technical Documentation Archive, 1997. http://www.omg.org/corba

[16] Microsoft Corporation,"Microsoft Component Services Server Operating System - A Technology Overview", August 1998

[17] Steve Deering, Allison Mankin and Lixia Zhangopen Research Problems on Scle and scalability", White Paper on Research Directions for the Next Generation Internet