Management Architecture Framework for Active Networks

Alden W. Jackson, James P.G. Sterbenz
BBN Technologies, Verizon *
Yechiam Yemini, Alexander V. Konstantinou
Columbia University †

30 June 2003

*This work was sponsored by the Defense Advanced Research Projects Agency issued by the AFRL under contract F30602-99-C-0131.

†This work was sponsored in part by the Defense Advanced Research Projects Agency contract DABT63-96-C-0088.

Contents

1 Introduction

Traditional network management provides management application tools at a Network Management System (NMS) for operations administrators to monitor and control element configurations, performance and failure modes. NMS manager applications access element agents who instrument monitoring and control functions. SNMP provides such GET/SET/Notify access by NMS managers to element agents by organizing the instrumentation in a global naming directory – the Management Information Base (MIB).

This traditional manager-agent architecture is based on several fundamental assumptions:

1. The task of management software is to provide tools for administrators who monitor and control the network through man-in-the-loop operations

2. Element configurations and network topology remain static and change infrequently

3. Management tasks – such as provisioning, configuration changes and problem management are handled in non real-time with operations

4. Elements play a passive role of exporting instrumentation access through local agents; the locus of control is with manager applications and operations staff. Elements are independent of each other, with administrators coordinating their configurations and correlating their behaviors

Active networks present fundamental new management challenges. The four basic assumptions above are no longer valid. An active network evolves dynamically as new elements are deployed, provisioned, configured, changed and deleted; dynamic changes are the norm rather than the exception. Both, element configurations and connectivity relationships can change on rapid time scales, not controllable through a man-in-the-loop administration paradigm. Elements can no longer play a passive role and function independently of their environment; instead they require autonomic capabilities to adapt and control their operating environment through coordination with other elements. Therefore, a major challenge in managing active ever-evolving networks is to extend traditional static/man-in-the-loop/passive network management paradigm to support a new dynamic/autonomic/active network management paradigm.

The challenges of dynamic/autonomic/active management permeate through the entire stack of current management architecture. We illustrate these challenges through examples. Consider an active application (AA) deployed dynamically in the network; it would be necessary to deploy with it a respective instrumentation MIB and manager applications to monitor and control the AA. These instrumentation and managers become active elements too, whose operations must be intimately coordinated with the dynamics of the AA. The SNMP model, furthermore, allocates manager functions (monitoring and control) to NMS applications with element agents functioning as passive instrumentation servers. With active networks AAs and EEs must monitor and control their environments and thus perform manager functions as well as export agent functions; this leads to a diffusion of manager/agent functions with elements, creating autonomic management model, in contrast with the sharper NMS/agent division. Furthermore, SNMP assumes that MIB schemas (SMI) are statically replicated at the NMS and at element agents to guide protocol access. The NMS is typically configured statically to include the SMI of all elements it must manage. With an active network the SMI of an AA MIB must be available to autonomic elements that need to adapt to, or control, the AA. This means that the SMI schema must be dynamically deployed and manipulated by multiple AA elements. SNMP's static replication model must be extended to a full repository model which supports shared access and dynamic manipulations of the very SMI schema and coordinated reconfiguration of the active instrumentation they represent.

Dynamics introduces significant new complexities to MIB design. For example, SNMP typically accumulates historical data through counters; these counters, furthermore, may be used to identify significant management events through thresholding. Suppose an AA is deployed, encounters a problem and aborts repeating this intermittent activations indefinitely. The MIB associated with the AA will be deleted with every such aborted activation and thus lose record of history. Yet to identify and analyze such intermittent dynamics it is necessary to accumulate a historical record of the instrumentation values. Therefore the AA MIB will need to provide persistent footprint and correlate it across multiple activations.

To summarize, managing active networks requires novel active autonomic management technologies that extend beyond the fundamental assumptions and paradigms of existing management frameworks. There could be two approaches to creation of such active autonomic management. One alternative would seek to construct a radically novel framework and mechanisms that

resolve the challenges. This would involve design of distributed repository structures that can support active autonomic management of dynamically evolving networks; mechanisms to deploy active management components and coordinate their monitoring and control activities with those of EEs and AA; and mechanisms for coordinated autonomic configuration, problem and performance management. All these require substantial research to develop effective solutions.

A second alternative, pursued by this paper, is to use an Occam's Razor design. Under this option existing frameworks and mechanisms are extended as minimally as possible to support a sufficient base to manage active networks. The goal is to provide the simplest working model leaving the broader questions and challenges of the first alternative to further research and future extensions. In what follows we describe such Occam's Razor design of an SNMP-based framework to support active autonomic management of active networks.

2 Assumptions and Requirements

A Managed Active NOde (MANO) is an active node that provides the base management capabilities described below.

2.1 Required capabilities

1. A MANO shall support SNMP-based monitoring and control of active nodes.
2. A MANO shall provide instrumentation to monitor and control the system and network resources of active node and of the EEs it executes.
3. A MANO shall enable EEs to provision node and network resources required for their execution.
4. A MANO shall protect management components against security attacks.

2.2 Optional capabilities

1. A MANO shall enable EEs to dynamically deploy MIBs with its SNMP agent.

2. A MANO shall enable EEs and AAs to manipulate MIBs safely and securely

3. A MANO shall provide an active management EE supporting dynamic delegation of management applications and their execution under local and remote controls.

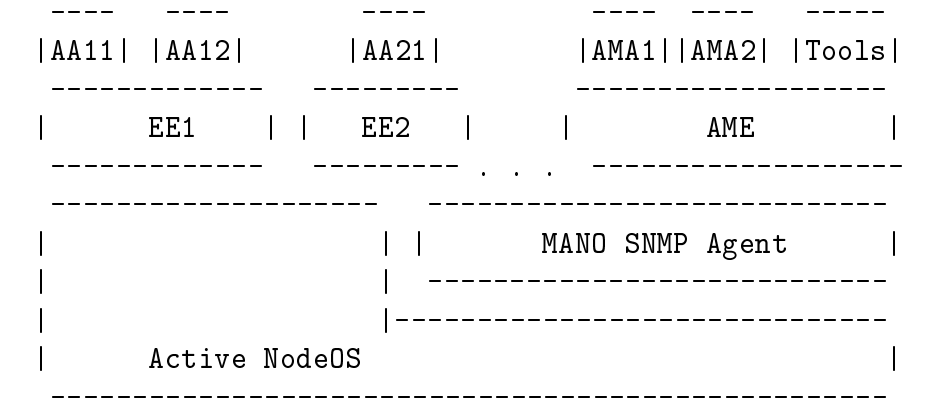
4. The management EE shall support standard management application tools to test liveness of links to neighboring nodes and the status of their EEs.

3 Architecture and Functions of A Managed Active Node

This section describes the base architecture, components and operations of a managed active node.

3.1 Base MANO Architecture

The overall architecture of a MANO is depicted in the figure below. A MANO contains an SNMP MANO Agent and an optional Active Management EE (AME) that executes active management applications (AMA).



The MANO Agent maintains MIBs to support an active management framework. It provides standard SNMP access to these MIBs by remote

managers as well as more extensive, safe and secure API for local access by EEs and by AME applications.

The MANO Agent incorporates several standard MIBs into a MANO MIB used to manage the active node environment. These MIBs include adapted forms of standard MIBs to instrument the node OS and resources (Host MIB); an ANet MIB to monitor and control links to neighboring ANet nodes as well as network resources that facilitate them; and a Registration MIB that supports dynamic registration of Active MIBs by EEs and establish protection features of their access. The Registration MIB provides the facility for dynamic deployment of MIBs and SMI schemas by EEs and their AAs.

The optional Active Management EE (AME) provides a standard EE to deploy and execute active management applications (AMAs). The AME provides a standard MIB, the AME MIB, to deploy monitor and control AMAs. These AMAs can access the MANO MIBs according to their protection capabilities and use these MIBs to monitor and control the operations and resources of the active node and its EEs. The AME supports certain universal tools to monitor and control the node, network and EEs operations. These tools, organized as AMAs shall include testing liveness of links to neighboring MANOs and of their EEs; tools to provision EEs, allocate resources to them and to change their configurations.

An EE could only monitor and control its allocation of node resources and its operations through the MANO Agent, either directly or as a side effect of invoking respective node OS functions. An EE may use the MANO Agent to register its MIB and enable local and external management applications to access them. Alternatively, an EE may execute its own SNMP agent (or any other management mechanisms) as an AA and bypass the MANO agent in supporting its management.

If the MANO incorporates the optional AME, then its EEs may use the AME to deploy and execute AMAs to manage its operations or those of its AAs.

3.2 Support SNMP Access

All MANO must support SNMP access to local instrumentation MIBs. In particular, a MANO needs to support a local SNMP agent (the MANO Agent) and SNMP/UDP/IP stack. A MANO is also assumed to incorporate standard SNMP security mechanisms (...LOL) to protect external accesses to the MANO agent.

Additionally, a MANO provides local API for secure access to all local MIBs. These API shall support full repository access to both SMI schema and their instrumentation data; protect the safety of the MIBs data through standard transaction interfaces; and protect the security of the MIBs data through standard access control.

3.3 Provide A MANO MIB

All MANO must support the MANO MIB. The MANO MIB shall include the Host MIB to instrument the MANO OS and EE processes, including appropriate extensions that permit identification of specific EE processes. The MANO MIB shall additionally include a common EE MIB to monitor the status and traffic of all EEs executing at the node. The MANO Agent shall provide a set of traps to indicate standard status/failure/activity conditions of EEs.

The MANO MIB shall also include an ANet MIB to monitor and control the local ANet topology and its mapping to underlying network resources. The ANet MIB shall incorporate instrumentation to provision, configure and monitor links to neighboring active nodes and allocate to them underlying network resources (e.g., QoS features). The MANO Agent shall provide a set of traps to indicate standard status/failure conditions of these links to neighboring active nodes and of the underlying network resources.

3.4 Support Active MIBs of EEs

All MANO must support active MIB (AMIBs) of EEs (and their AAs) as follows. An EE shall be able to deploy, activate, deactivate and delete MIBs and respective traps, as well as control their protections. The following MANO facilities are required.

1. There shall be an AMIB Registration MIB, organizing information on all AMIBs registered by EEs with the MANO Agent.

2. The Registration MIB shall incorporate for each EE the following data: all AMIBs registered by the EE; for each AMIB its SMI definitions, its binding to instrumentation executed by the EE, traps associated with it, its status and its access protection specifications. Notice that multiple instances of an EE will have independent Registration MIB data.

3. The MANO Agent shall instrument the Registration MIB to enable EEs to register AMIBs, change their status, instrumentation bindings and protection structures and delete them as needed. The MANO Agent shall also terminate and delete all AMIBs associated with an EE upon termination of such EE.

The Registration MIB thus provides a limited repository infrastructure that permits dynamic deployment of MIBs and SMIs and their shared protected access and manipulations by management applications, EEs and AAs. It should be recognized, however, that these capabilities are bounded by intrinsic limitations of SNMP in supporting safe and secure transaction semantics. It is implicitly assumed here that such broader repository functions can be provided using more advanced mechanisms beyond the minimal scope of this design.

3.5 Optional Support of Active Management EE

MANO could optionally support an Active Management EE (AME). The AME shall provide a common environment to deploy active management applications (AMA) to support autonomic self-managing features of a MANO. For example, an EE could deploy a management application that provisions links to neighboring active nodes, replicate the EE at selected neighbors and allocates node and network resources to this EE overlay. For another example an EE could deploy a management application that monitors its operations, automatically reboots it upon failure and instruments log of such failures in a respective MIB to enable other management applications to detect, diagnose and recover from intermittent failures or report bugs. Similarly, an AA for content-distribution service could deploy a performance management application that monitors traffic patterns and underlying topology status and reconfigures the AA to respond to emergent changes in these.

The AME shall incorporate mechanisms to deploy (delegate) AMA. It shall enable AMAs to access and manipulate MIBs of the MANO Agent, according to their security permissions. It shall instrument an AME MIB to monitor and control AMAs and register it with the MANO Agent. The AME MIB shall also provide standard traps to notify significant status events of AMAs. The AME MIB shall enable EEs to deploy, execute, monitor and control AMAs associated with their operations.

4 Tools

Support for a small number of management tools is needed to encourage the development of a rich active network infrastructure.

4.1 Discovery

Many AAs need knowledge of the network topology for correct operation. Others need the topology information for deployment or optimization.

Management has three areas of interest in topology discovery:

1. The topology of MANOs.
2. The topology of particular EEs running on MANOs
3. The topology of particular active applications deployed on a set of EEs

The information on the topology of active nodes resides in the ANet MIB. As stated in Section 3.3, the ANet MIB incorporates instrumentation to to provision, configure and monitor links to neighboring active nodes.

The development and maintenance of the neighbor information for a particular EE on a MANO is optional. However, EE's AMIB can be designed to maintain neighbor information. The AMIB binds the EE-specific instrumentation to status variables. Upon EE invocation, the AMIB Registration MIB, registers the AMIB for this EE with the MANO MIB. Correspondingly, the when links to "neighbors" of this EE are discovered, they can be identified as entries in the AMIB.

The development and maintenance of the topology of active applications deployed on a set of EEs in the active network is the responsibilty of the the EE or application in question. The manangement infrastructre supports the identification of active nodes and their corresponding connectivity, which can be used by the EE or application to support topology at this level. It is an architectural decsion as to whether the EE's AMIB is designed to track AA-specific information, or if the AMIB parallels the MANO MIB and supports the registration of AA AMIBs.

In general, topology discovery is very difficult in existing networks. While this document cannot address the problem of layer-2 topology discovery, the MIBs must provide sufficient information to perform end-to-end, multi-layer topology discovery, assuming that information on the layers is available.

4.2 Reachability

In order to establish communication to an EE or between EEs, reachability has to be determined. Management has two areas of interest with respect to reachability:

1. KEEP-ALIVE to check EE link availability (neighbor)
2. PING of EE through IP (management channel)

Both of the above are similar to the ICMP echo service widely used in IP. EEs should support a KEEP-ALIVE service that can determine if the link between neighboring similar EEs is operating. Since the "link" is an EE-derived notion and may cover multiple hops through conventional network nodes, the KEEP-ALIVE service should have the EE perform some work to indicate that the peer is not only reachable (with respect to the underlying network), but also capable of communication and/or computation.

Earlier we stated that the basic management channel to an active node was via IP. IP services alone can determine if an IP path to the active node exists. However, IP services are not able to determine if a particular EE is running on the active node. EEs should support an interface through the management facilities within the NodeOS to determine EE liveness. Thus a query can be made from the NodeOS of an active node to "PING" an EE to determine its liveness.

5 Security

Several of the classical communication threats to network protocols are applicable to the network management problem. In addition to the threats to communications, there exist threats that occur by executing code contained in AAs, as detailed in [1]. The correct operation of the network requires that individual routers are not subverted from forwarding packets correctly. Thus, protecting the correct operation of the router and its configuration is also a goal of these requirements. This section discusses the principal threats to an Active Network Management Architecture.

5.1 Threats

The principal communication threats to a robust management model are:

Modification: the danger that some unauthorized entity may alter in-transit messages generated on behalf of an authorized user,

Masquerade: the danger that operations not authorized for some user may be attempted by assuming the identity of another user that has the appropriate authorizations, and

Disclosure: the danger of eavesdropping on the exchanges between managed nodes and a management station(s). Protecting against this threat may be required as a matter of local policy.

Note that the above can be countered by integrity, authentication, authorization, and confidentiality services.

The principal operating environment threats to a robust management model are:

Unlimited consumption: the danger that a program can consume node resources without bound,

Unlimited access: the danger that an unauthorized program can access or effect sensitive areas, and

Unsafe evaluation: the danger that a fault during program execution can cause harm to the node evaluating the program.

Reference monitors or low-level instrumentation in the NodeOS can be used to protect against unlimited consumption. The NodeOS architecture specifies that enforcement mechansms should exist control running programs. Authorization policy, including access control mechanisms, if implemeted in the NodeOS, as suggested by the Security architecture, should prevent unbounded access by unauthorized entiies. Dangers from unsafe evaluation can be addressed by various methods, including evaluating the program in a tightly controlled sandbox to verifing that a proof carried with the program meets the security/resource policies of this node. The nature of the protection and its implementation is a function of both the NodeOS and the EE implememtations.

5.2 Needed Services in the NodeOS

The Security architecture provides strong arguements for placing the responsibilty for the authentication of packets in the NodeOS, which are briefly reiterated here:

- The security functions are of common use to all EEs (including and especially those used for management functions)
- The NodeOS had resource of its own to protect and thus requires the functionality
- Some types of processing or communications are difficult to protect if the authentication support was in an EE. For example, cut-through channels are designed to avoid EE processing.

The management architecture should leverage the availabilty of the following components from the Security architecture:

Cryptography: providing integrity, authentication and key management functions

Credential: a system to create, store, retrieve, disseminate and revoke credentials

Policy: a datebase to store policy statements and an engine to evaluate policy during enforcement

Enforcement: binds security context to code execution, such that context is available to authorization functions but not the code.

References

[1] S. Murphy, ed., "Security Architecture for Active Networks," AN draft, AN Security Working Group, Nov. 2001.