An active, policy based management architecture for multi-service networks

I.W. Marshall, J.Hardwicke, H.Gharib, M.Fisher and Paul Mckee

BT Labs, Adastral Park, Martlesham Heath Ipswich IP5 3RE

Abstract

Future multiservice networks will be extremely large and complex. Novel management solutions will be required to keep the management costs reasonable. Active networking enables management to be delegated to network users as a large set of independent small scale management systems. A novel architecture for an active network based management solution for multiservice networking is presented.

Introduction

Usage of multi-service networks is increasing rapidly as global take-up of the heterogeneous Internet increases. At the same time the number of services offered is growing exponentially, along with the associated management costs. To minimise the cost and complexity of this large distributed system, novel approaches to system and service management are required. Previous proposals have typically involved some attempt to reduce the interdependency of system components and enable management actions to be undertaken purely on the basis of the state of individual components. One approach is ‘Management by Delegation’, as proposed in [1]. Here, the management of applications is delegated to the servers where they execute, and thus the processing load is shifted onto the application server. However, such a system retains a centralised model for information handling and delegation, and appears unlikely to scale to the degree envisaged.

The use of mobile agents has been proposed as an alternative solution [2,3]. The agents are migratory programs that move through the system collecting data. Rather than communicating data to a central source the agents may act autonomously, either alone, or in collaboration with nearby agents to resolve system problems. However the agents tend to be large and complex, and their migration through the system imposes considerable overhead, reducing performance.

Active networks [4] aims to enable network clients to easily add new services. Users of an active network can supply the programmes and policies required for their custom services, in transport packets alongside their data. This increases the available network functions dramatically, and at the same time reduces network operating costs. The latter benefit is achieved by effectively delegating expensive management functions associated with the client defined services, to the clients. However, to realise the benefits, a more flexible management system is required. In particular, there should be no restriction on who can delegate to what, nor on the location of the ‘delegator’. One must also provide a common information model for server and network resources ensuring management information can be passed around the system in a scalable manner.

We have designed and partially implemented an active management solution for multi-service networks based on role-driven policies [5,6] and Application Layer Active Networking (ALAN) [7]. Our system provides an extensible monitoring and configuration service that enables users to specify their configuration, monitoring and notification requirements to network devices using policies. Each policy specifies a subject (the policy interpreter), a target list (the objects to be changed if the policy is activated), an action list (the things to be done to each target) and the authorisation code, id and reply address of the originator. The policies are named using a universal policy name, which is also part of the policy. The names currently take the form *upn:originator id.subject.target_list.last_modified_time* and are likely to be globally unique. The policies are multicast to relevant hosts, where they are received by a management agent, and stored in a local policy store if the appropriate key is present (i.e. a key associated with a role authorised to supply policies to the target device). The management agent has an extensible table of authorisation policies to enable this decision. Roles are allocated by storing the appropriate key on the users smart card. A policy addressed to the management agent can consist of a number of component policies, each of which specify the subject (normally an object oriented program) intended to use it as part of their rule-base. The policy store has a table of policies for each registered subject and the local agent will store the component policies in the appropriate parts of the database.

Our approach avoids many information handling problems by using a lightweight scalable mechanism for information transfer. The Information Management System [8] consists of a hierarchy of '*store and forward*' information stores, with events being classified by their propagation characteristics and storage duration. Two types of these information stores are used, their selection depending on the complexity of querying required against storage availability. While simple but fast stores offer a load balancing and traffic controlling function, more complex stores permit management information analysis. Our solution is flexible and able to accommodate the requirements posed by the active networks far more easily than current solutions such as Tivoli that assume unitary administration authorities.

Further details

Figure 1 shows the management facilities required at each host. The management agent receives policies from any managers who send them, and stores them in the policy store if authorised by the managers role. It also informs the subject(s) and associated management facilities of the update to the store and generates an event recording the change in the event log. The subject will then enact the policy. For example the policy may specify the subject must record each usage instance for a particular interface that it supports. The default target is the local event log so the subject must forward the usage data to the event logger. The event logger will generate an event from the base data, adding a timestamp, a sequence number, an event class, a time to live, the source object and the generating policy name(s). The event logger must also inform the filter service that the event has occurred if it has a policy requesting it to do so for events matching the one it has just created. The filter service has a set of policies regarding notification of events to interested managers, e.g. is encryption required, which events should be sent (thresholding), should they be sent real time or in batches, is interval detection or combination detection required, etc. These policies can reference interfaces offered by the management facility or additional objects supplied by the originating manager. Once any preprocessing has been performed the filter service then forwards the event to the despatch service. The despatch service has policies indicating the QoS required for transmitting the event, and invokes the appropriate protocol wrapper and identifies the destination address list before sending the notification.

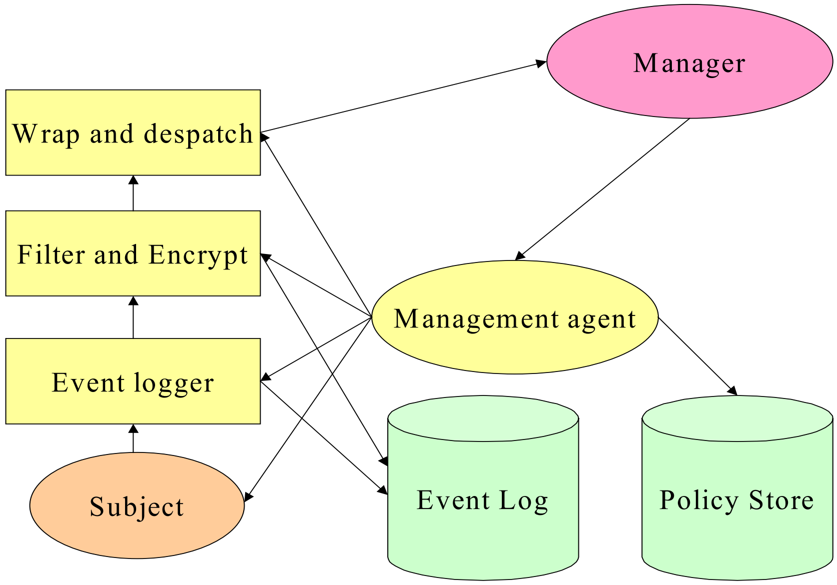

Figure 1 - Management services schematic for a single host

The address list is identified using a name server (not shown in the figure). The name server maintains a list of subjects, targets, policies, and manager ids registered at the local node. Each list resolves to the address(es) of the entities. In the case of managers the address could be a multicast address the manager obtained from the information management system.

The current status of this system is that we have implemented the information distribution system and the policy store, but have not yet implemented the management agent. We anticipate an initial implementation will be available in the next few months enabling the full architecture to be experimentally validated. In the meantime we are confident we have created a system that offers a significant enhancement over the somewhat rigid SNMP based systems currently available, that is much better suited to the needs of a customer driven, active multiservice network.

References

[1] Yemini Y., Goldszmidt G., Yemini S., "Network Management by Delegation", Proceedings of the 2nd International Symposium on Integrated Network Management, April 1991.

[2] Gavalas D., Greenwood D., Ghanbari M., O'Mahony M., " An Infrastructure for Distributed and Dynamic Network Management based on Mobile Agent Technology", ICC 1999.

[3] Ku H., Luderer G., Subbiah B., "An Inteligent Mobile Agent Framework for Distributed Network Management", Proc. IEEE Global Telecommunications Conference (Globecom '97). Pp160-164, 1997.

[4] Marshall I., Cowan J., Crowcroft J., Fry M., Ghosh A., Hutchinson D., Parrish D., Phillips I., Sloman M., Waddington D., "Alpine - Application Level Programmable Inter-Network Environment", BT Technology Journal, April 1997.

[5] Sloman M., "Policy Driven Management for Distributed Systems", Plenum press Journal of Network and Systems Management, Plenum Press

[6] Lupu E., Sloman M., " A Policy-based Role Object Model", Proceedings of the 1st IEEE Enterprise Distributed Object Computing Workshop (EDOC '97).

[7] Fry M., Ghosh A., " Application Layer Active Networking", Computer Networks and ISDN systems, 31, May 1999.

[8] Bates J., Bacon J., Moody K., and. Spiteri M., “Using Events for the Scalable Federation of Heterogeneous Components”, *Proceedings of 8th ACM SIGOPS European Workshop*, Sintra, Portugal. September 1998.