Towards Policy-Based Management of Active Networks

Alvin Tan (atan@ee.ucl.ac.uk) and Alex Galis (a.galis@ee.ucl.ac.uk)

University College London

Abstract: This paper represents on-going efforts within the IST project, FAIN (Future Active IP Networks). It begins with an introductory section, which briefly addresses the main challenges faced in policy-based network management (PBNM) initiatives. The FAIN network management approach is also presented. It further captures requirements in PBNM by means of the characteristics and methods in adapting its implementation to active nodes. The exposition of active node functionalities is presented using the management functional areas as represented by FCAPS.

1 Introduction

In comparison with previous traditional network management approaches [6], the policy-based approach offers a more flexible and customisable management solution, thus allowing network elements to be configured 'on-the-fly' per application, per customer. Originally developed for a LAN-like network environment, this flexibility comes with a cost because this approach renders the need to address security and scalability issues. The semantic of policies presents another challenge in current policy efforts. The policies that can be defined are limited by current information model. Currently, this includes only classes for the representation of policy conditions based only on time and on packet headers. The current policy-based network management (PBNM) architecture is able to address fairly limited domain issues, i.e., those that can be translated to fixed configuration settings. For example, quality-of-service (QoS) issues often need complex interactions between relevant network management components and these complex interactions cannot be easily implemented in the current PBNM architecture. Moreover, according to the policy framework, policies are defined or modified by an administrative tool and the intervention of the administrator is always required. Active networks [1] may resolve many of the problems inherent in current PBNM technology. It allows the dynamic enhancement of the management architecture, the introduction of new application as well as automation of the network management tasks. The Policy Based Active Network Element Management (PBANEM) architecture described in the following chapters demonstrates how the policy-based approach is adapted into an active node framework.

2 FAIN Active Node Reference Model

The following are components of the FAIN Active Node Reference Model.

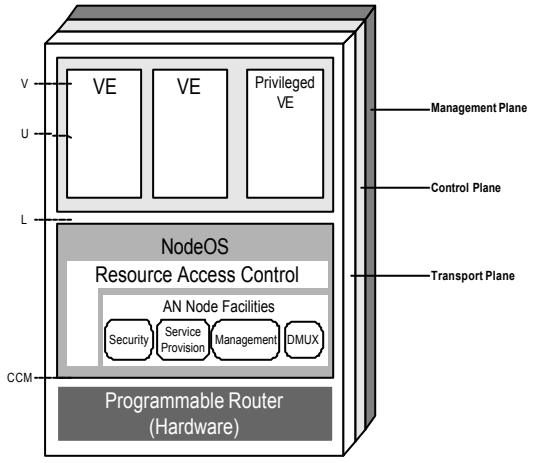

Figure 1: The FAIN reference architecture

Execution Environments (EE): Services are what the EEs have to offer, usually in the form of static interfaces. Services may well be extensible in the sense that EEs offer another interface (service) through which the service under consideration can be extended. It may be thought as extending the NodeOS 'upwards' into user space.

Virtual Environment (VE): A VE provides a place where services may be instantiated and used by a community of users and stay isolated from other communities. Within a VE many technologies may be used to implement and/or instantiate a service.

Node Operating System (NodeOS): The most important component of the reference architecture is the NodeOS [8]. It provides basic functions from which EEs build the abstractions that make up network APIs. It offers all those facilities that are necessary to keep all the other components together, whilst isolating EEs from the details of resource management and the existence of other EEs.

Node External Interfaces / Network APIs components: Application Programming Interface (API) provides mechanism by which objects transparently make requests to and receive responses from other objects on different platforms in heterogeneous distributed environments like Active Networks. The API is object-oriented and consists of several categories of interfaces as follows: Service Interfaces offer applications access to a range of network capabilities; Framework Interfaces provide 'surround' capabilities necessary for the Service Interfaces to be open, secure, resilient and manageable; Administrative interfaces are to support administrative functions within the enterprise and to permit the supply of services by third party vendors. A working version of the APIs will produced as IDL specifications.

3 FAIN Network Management Approach

3.1 Motivation

It is expected that a management execution environment (EE) will have privileged access to the node resources through the P1520 interface [9]. The idea is to create proxies (an object-oriented architecture is advantageous for this

purpose) to the objects that represent the abstractions of resources. Such proxies would consult a credentials database before granting access or rejecting the requests. In any case, the same interface should be presented to every EE. In FAIN, the service logic for the policy-based provisioning of a sample service (e.g. virtual private networks or QoS policies, depending on progress in the IETF) will be implemented on top of the node API. The following benefits of active networks to network management are expected:

Flexibility: With active networks, not only can policies be expressed as data structures (which are limited by the constraints imposed by standardised database schemas), but they can also be expressed quite flexibly as active code. Policies expressed as active code will be able to have more inherent intelligence than 'plain vanilla' policies.

Automation of management tasks: Active networking technology will allow the PBANEM system to dynamically extend its management functionality by downloading new components. An important goal of any management architecture is to automate tasks as much as possible. Active networks will allow node self-reconfiguration due to changing network conditions.

Delegation: Network operators see the delegation of management tasks to customers as an important property of the management architecture, since it will dramatically reduce their network management costs. Using active networks a secure and effective method to delegate management tasks to customers (who will gain more control over its contracted resources) can be created.

Application-specific management: The ideal network and node configuration for each application might be different, and might change depending on network conditions. Active networks provide the technology to allow applications to install its own policies to configure node and network resources in a secure way, hence achieving optimum usage of resources in different network or node conditions.

3.2 FAIN Policy Based Active Network Element Management (PBANEM)

The use of policies for network management has recently been introduced to the Internet community. However, for the deployment of PBNM systems in the Internet, a standardis ation process is required to ensure the interoperability between equipment from different vendors and PBNM systems from different developers. Both the Internet Engineering Task Force (IETF) and the Distributed Management Task Force (DMTF) are currently working for the definition of standards for Policy Based Network Management. The management approach in the FAIN project is based on policies. We envisage that the management of the active network [7] will require the following components:

- **Policies:** Design of management policies required to manage the active nodes and network.
- **Management Components in the Active Nodes:** Design of management components for the active nodes, which will execute policies within an active node and which will monitor the node resources usage. The execution of policies means mapping target policies into node resource configurations.
- **Management Nodes:** A set of management nodes that will mechanisms to enable network administrators to manage the active networks as a whole, including network policies set-up and processing.

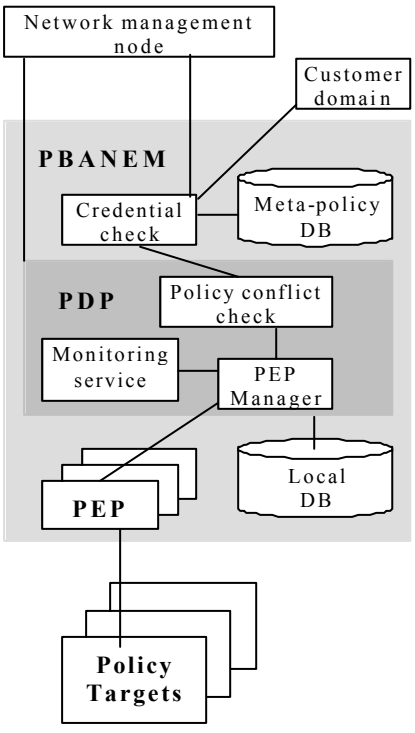

Figure 2 Policy Based Active Network Element Management

A two-tiered management framework is envisaged: the network level, and the network element level. In FAIN there will be a small number, maybe just one, of network managers per domain that will implement the network level management functionality. While, the active network element manager will implement the network element level management functionality. We will have one element manager per active network element and this will be explores in the subsequent sections. As in TMN [4], the element management system (EMS) manages one or more of a specific type of telecommunications network element (NE). Typically, the EMS manages the functions and capabilities within each NE but does not manage the traffic between different NEs in the network. To support management of the traffic between itself and other NEs, the EMS communicates upward to higher-level network management systems (NMS).

The architecture (Figure 2) is loosely based upon the IETF Policy Framework [5], whereby the main components of the policy framework are the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP) [2]. The PDP addresses two main aspects of PBNM. It takes charge of the retrieval of the policies coming from the NMS or from the active applications, proceeding to their distribution to the appropriate PEPs. Secondly, it determines the policies to be applied in every applicable scenario depending on its knowledge of the node status. The PEP is responsible for enforcing these policies on the appropriate managed resources. Local conflict detection and resolution

[3] is, in fact, one of the major functionalities of the PDP. The conflict check component in the PDP is an essential condition since local conflict detection requires understanding the semantics of the policies (conditions and actions). The knowledge that the PDP has about the semantic of policies, in order to realise the conflict detection, can be dynamically enhanced by downloading extensions to the conflict check component along with the PEP that will enforce this policy or with the policy itself.

In order to tackle potential problems not known at the time of PDP design (e.g., those coming from active applications), we assume the use an active approach. The active packets containing the meta-policies (data) will also contain the condition or action interpreters (code) required to execute the policy, which are not in the system yet. These interpreters will be treated as the pieces or 'building blocks' of the policy interpreter. This way, the PDP can acquire knowledge about the policy classes used by the PEP. Using this scheme, the same PDP type can be used for managing different types of enforcement clients. The computational resources of the node are the processing power (CPU cycles), the memory and the disk storage. An active node may have multiple EEs. These EEs are competing for the resources of the node and therefore the administrator must set up the corresponding policies to partition the computational resources of the node as needed. There is a need to assign packet flows to EEs. The PBANEM system is also responsible for the creation, deletion and modification of EEs. When the creation of a new EE is requested, the PBNM system should check existing policies, to see if the person that made the request has the necessary privileges and if there are sufficient resources in the node for this EE. There should also be priorities for different EEs, to ensure that the most critical EEs always have enough resources to execute. These priorities also depend on Service Level Agreements (SLAs) made between the network providers and the consumers.

4 Management Functional Areas

4.1 Configuration Management

The configuration of the system can be done in three ways: provisioning, signalling and self-adaptation. In the provisioning scenario, the network administrator defines policies using a tool application and sends them to the PBANEM System. In the signalling scenario, resources are configured using a signalling protocol. The signalling requests cannot be directly resolved by the PEP, because this component does not have the capability to make decisions on its own. In the self-adaptation scenario, the PBANEM system must be able to automatically set new policies, depending on the status of the managed nodes, without requiring the manual intervention of the administrator.

The tasks of creating, deleting and modifying an execution environment within the active node raises the necessity of the element management system to access certain resources of the active node. It should be able to: assign computing resources (CPU cycles, memory, etc.) to the execution environment; assign flows to EEs; and, install and remove code within the EE. The reservation request for both computing resources and forwarding resources can come from the management system or from the active node (signalling protocol). In order to support this second alternative the active node interface should be able to support the ability of sending reservation request to the PBANEM system.

4.2 Fault Management

Active networks can enhance critical resource fault management (Figure 4 below) since they provide the intelligence to foresee the possible faults and take rapid decisions as required. We distinguish two main fault types that should be considered when defining the management architecture: critical resource faults and non-critical resource faults. The former limits the node operability, causing node collapses and therefore have a severe impact on network

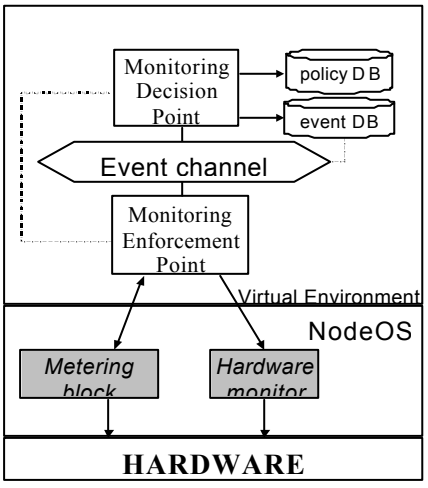

Figure 3: Service monitoring Architecture

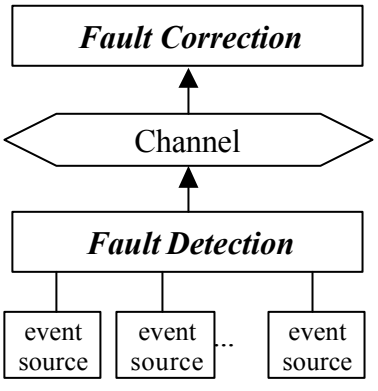

Figure 4: Local fault management system blocks

behaviour. The non-critical resource faults do not cause a serious damage in node working, hence the local management system is able to proceed with the control functions and autonomously recover from the failure. For example, these types of faults could lead to a performance failure. Several policy types could be defined in such a case to give priority

to certain activities, normally those that are essential to avoid the node collapse. In both cases, it is necessary to implement the appropriate mechanisms for fault notification and local decision dissemination. In this way it is possible to modify the global policies that could be affected as a consequence of the local node's anomalous behaviour.

4.3 Performance Management

The performance management functionality monitors a number of active node characteristics or resources via its interface in order to obtain statistical information. The most important information that should be available is information about node computing resources status like CPU use percentage, use memory, free memory, and available hard disk. Moreover, this information should be given in general and per flow, number of threads in the node, number of threads per execution environment, etc. The Monitoring Service component (Figure 3 above) of the FAIN PBANEM system is responsible for the continuous monitoring of active node resources, which are included in the conditions of system policies. The functionality of the Monitoring Service can be split into two parts. First the Monitoring Service will enable the PEP Manager to register the conditions of the system policies. Then the Monitoring Service instantiates the corresponding metering blocks, which will monitor the registered conditions. When these conditions become true, the Monitoring Service will send an event to the PEP Manager. To implement this functionality, the basic interface offered by the Monitoring Service will support an operation for the registration of a new condition. The parameters of this operation should be the condition expression and the IDs of the PEPs that will have to be monitored.

4.4 Accounting Management

The information obtained from the monitoring of node resources listed in the previous section, as well as the Configuration Management information that the management system has for each component will be sufficient for realising the accounting management tasks within the PBANEM system.

4.5 Security Management

From the security management point of view, we have to consider two aspects. Firstly, the security of the active node itself is in question. In order to avoid that a non-authorised person manipulates node resources, the privileged interface should only be accessible from the exterior for the management system. Secondly, another important security requirement is that the interface should be able to provide authentication information of the requesting principal in whose behalf a reservation request is made from the active node to the management system. This authentication information will avoid the reservation of resources to a principal who does not have the appropriate reservation rights. In general, all policies or requests arriving at the management system should provide a credential allowing to that principal the realisation of the requested management tasks.

5 Conclusion

The main task in FAIN is to develop and validate open, flexible, programmable and dependable network architecture based on novel active node concepts. It is imperative that a management node is developed in a parallel effort to accommodate unique requirements that come with managing an active node. The policy-based approach is adopted and there will be extensive on-going initiatives within the project to investigate and develop pragmatic yet resilient solution for the management architecture. Validation exercises will be conducted by realising applications scenarios, e.g., VPN provisioning with QoS guarantees.

6 References

[1] D. Tennenhouse, D. Wetherall, **'Towards an active network architecture'**, Computer Communications Review, 26, 2 (1996), pp 5-18

[2] R. Yavatkar, et al, **'A Framework for Policy-based Admission Control'**, RFC 2753, January 2000

[3] E. Lupu, M. Sloman, **'Conflict Analysis for Management Policies'**, Proceedings of the 5th International Symposium on Integrated Network Management IM'97, San Diego, Chapman & Hall, May 1997

[4] http://www.itu.int/TMN/

[5] M. Stevens, et al., **'Policy Framework'**, Internet Draft, March 2000

[6] Galis, A. (ed.), **'Multi-Domain Communication Management'**, pp. 1-419 and Appendices, pp. 422 –1160- in CD-ROM; CRC Press LLC, Boca Raton, Florida, USA, ISBN 0-8493-0587-X, July 2000

[7] Alex Galis, Alvin Tan, Joan Serrat, Yiannis Nikolakis, Julio Vivero, Spyros Denazis, Juan Luis Manas, Jan H Laarhuis, **'Policy-based Network Management for Active Networks'**, IEEE ICT 2001 Conference proceedings, Bucharest, Romania, 4-7 June 2001

[8] AN Node OS Working Group, **'NodeOS Interface Specification'**, January 2000

[9] J. Biswas, J. Vicente, M. Kounavis, D. Villela, M. Lerner, S. Yoshizawa, S. Denazis, **'Proposal for IP L-Interface Architecture'**, IP Subworking Group IEEE P1520