Policy-Based Management with Active Networks

K.L. Eddie Law, Kason Wong
University of Toronto

Abstract: A Policy-Based Management (PBM) system from IETF is responsible for resolving and enforcing policy rules in both the outsourcing and provisioning models so as to realize end-to-end Quality of Service (QoS) for all network connections. In this paper, we propose a novel architecture of the bandwidth brokers using active network technology. Active network paradigm empowers network nodes with the ability to manipulate data and program code as needed. In our proposed architecture, the Active Bandwidth Broker (ABB) framework distributes the policy control workload through the active nodes inside a policy domain. The Policy Decision Point in PBM is decomposed into two types of active application agents for traffic reduction and expedited decision processing. Moreover, the decision making agent is a mobile active program code that can move when the local network condition deteriorates. In order to verify the system performance of this proposed design, a working network prototype was constructed to demonstrate that the decision point can move to a comparatively low traffic region. As a consequence, the ABB system improves the overall performance when it is compared to PBM.

Key words: Policy-Based Management, Active Networks, Common Open Policy Service protocol, Policy rules, Quality of Services, Active Bandwidth Brokers, Policy Decision Point, Policy Enforcement Point

1. INTRODUCTION

Internet has evolved into a unified platform for serving traffic with different characteristics, for example, voice, video and data. Different Quality of Service (QoS) classes have been developed and they allow Internet Service Providers to provide different service classes with different performance guarantees.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35620-4_43
D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility*

With the availability of these multiple service classes on the Internet in future, proper control of network elements is desirable in order to provide predictable and controllable performance for different applications with different QoS requirements. Moreover, there exists many different access technologies, for example, the cable modem, digital subscriber line (DSL), wireless short message system (SMS). Each of them may require the designs and implementations of different qualitatively diverse control protocols and system models on the Internet. Hence these designs may further increase burden of the administrative system on the control and management of a network domain.

In response to this situation, Internet Engineering Task Force (IETF) has defined mechanisms that automate and ease QoS configurations within a policy domain. Policy-Based Management (PBM) [1] architecture is the proposed reference design; and the Common Open Policy Service (COPS) [2] is the enabling protocol that facilitates outsourcing and provisioning communications in the PBM system.

In this paper, we propose an experimental works on designing a policy control architecture using active network technology. It is known as Active Bandwidth Broker (ABB) framework. Active networks allow program code and data to move within a policy domain. Moreover, the decision point in the ABB may move according to certain design criteria. Therefore, there are two system control designs with the ABB model: (1) the number of decision points in a policy domain, and (2) the locations of the decision points. Theoretically, we like to move the decision point to a global optimal location such that all network traffic including policy requests and replies can be evenly distributed within a policy domain. However, there are several foreseeable problems. One obvious issue is that the decision point may oscillate and move to different locations rapidly. This oscillation situation will hamper the system performance and, as a result, it is undesirable. Moreover, it is difficult to measure network traffic in real-time if there are numerous devices to control in a large policy domain. Since it is a time-consuming process to implement the proposed ABB system model, we focus on finding a better placement for the decision point in our constructed prototype, and its design will be discussed in detail in this paper.

2. POLICY-BASED MANAGEMENT

The ultimate goal of the Policy-Based Management [1] design framework is to provide end-to-end policy enforcement across the Internet according to certain preset management rules. It shall include inter-domain negotiations among multiple decision points in neighboring domains for

connection establishments. There are several protocols designed for PBM in IETF. Among them, the COPS protocol provides a single interface for users, applications and network administrators. It defines two functioning entities in the system, the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP).

There are several operating models for the COPS protocols. They can be classified into outsourcing and provisioning models. These include the COPS, COPS-PR [3] and COPS-RSVP [4] protocols. In this paper, the basic outsourcing COPS model will be outlined and used to elaborate the ABB design. For the COPS outsourcing model, a connection request from an end-user or neighbor domain may reach an edge router (ER) or a boundary router (BR) correspondingly. Each of these routers functions as the PEP entity and it generates a COPS request message to the PDP in the policy domain (PD). Subsequently, the PDP asks for the related policy rules from a policy repository and makes a final decision rule for the PEP at the edge router to enforce the made decision policy. On the other hand, the COPS-PR [3] allows policy provisioning from a system administrator to enforce policy rules on network routers directly. Typically, this design is applicable to networks that provide Differentiated Services [5]. While the COPS-RSVP [4] provides extension for enabling policy control on networks that offers Integrated Services with the Resource reSerVation Protocol (RSVP) [6].

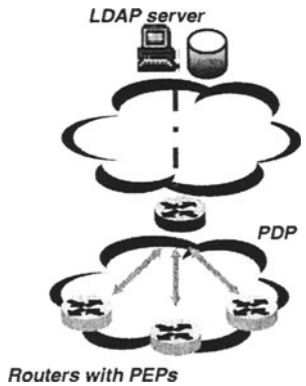


Figure 1. The PBM Model from IETF.

The reference policy framework [1] from IETF has a centralized policy manager that operates as PDP, and the policy repository is stored at a Lightweight Directory Access Protocol (LDAP) [7] server. Bandwidth Broker (BB) is an alternative name for the policy manager. The BB was initially used for setting Differentiated Services. In order to make a system adaptive, the BB can monitor network utilization and update current network

status within its policy domain. Hence, BB can help to calculate future usage of the network resources.

According to the retrieved or newly set policy rules from the policy repository, PDP delivers a policy decision to PEP where the policy action is enforced. This signaling mechanism assures that all traffic using the same path and service class can deliver information with the same QoS specifications. A basic structure of the policy-based management architecture with the COPS model is depicted in Figure 1.

The current PBM design is a centralized design. The Policy Information Base can get complicated and it is difficult to assume the PBM can work for a large-sized policy domain because the extra control mechanisms from COPS can introduce additional cost in network delay, traffic and system configuration of network devices. Therefore, the existing PBM design from IETF exhibits multiple noticeable problems in the large-scale networks, i.e., the system scalability and ease of configuration. In a large policy domain, there may have a lot of network routers (PEPs). We may then need to deploy more than one PDP in order to share the heavy workload. In that situation, the policy rules for a specific PEP may have changed and it may consequently affect the decision making that should be made for other PEPs in the policy domain. Therefore, it is a necessity to provide coordination between multiple PDPs. Moreover, its centralized design has system design problems with scalability issues; it may further produce undesirably low system performance for this two-tier PBM architecture.

3. ACTIVE BANDWIDTH BROKERS

In this section, we introduce the concept of active networking [8] and how it delivers a new conceptual design on active networking agents for the policy-based management. Active networks provide flexibility that allows users and administrators to send programs and data into the networks. Each active node may perform operations on received data sets based on their associated programming designs. Similar to all computing systems, active networks platform can be considered as a two-layer computing architecture. The part that functions like operating system is known as Execution Environment (EE). An application runs on top of EE, and this layer is known as Active Application (AA) layer.

In this section, we present a novel design for the Policy-Based Management with mobile agents operating in active networks. The resulting proposed design is an automatically re-configurable Policy-Based Management system. We called this design the Active Bandwidth Broker (ABB) framework.

In ABB, the policy decision-making agents can move freely within the networks. With this mobile capability, the proposed ABB architecture can achieve a more dynamic, reliable and scalable design. The focus of the paper is to develop and verify the ABB architecture. The ASP [9] is selected for building an ABB prototype because it supports two important functions: the dynamic class loading and active code caching. Moreover, ASP can be used to support complicated network control protocols for it supports several features that are not being found in other EEs at the moment, for example, the program code sharing, dynamic class binding and soft-state storage.

There are two system parameters in the ABB model: the number of decision points in a policy domain, and the locations of the decision points. However, they will be not discussed in detail in this paper. In the following, the partitioning of PDP in ABB will be discussed.

3.1 P-ABB and S-ABBs

In ABB design framework, the decision-making function of PDP will be programmed as the only mobile active application. Other functionalities of the PDP can thus be distributed to other active nodes in the domain. Thus it is natural to partition PDP into two different entities for the ABB, and they are the:

1. Primary – Active Bandwidth Broker (P-ABB), and
2. Secondary – Active Bandwidth Broker (S-ABB).

The Primary ABB (P-ABB) is designed to perform all required functionalities of PDP. To simplify the system complexity, P-ABB is the only entity that is responsible for determining the final policy for execution and sending the COPS decision message to a router for policy enforcement. For the prototype, we preset several system operations and entities in order to expedite the implementation process. For example, the P-ABB always keeps a management repository that saves all enforced policy requests and rules. The design of this part may be different if multiple P-ABBs are allowed to exist in a domain. P-ABB actively monitors the network traffic and always registers its location with a location server. The role of the location server is quite simple. It keeps the location information of the P-ABB and regularly checks its status. Since we focus on the mobility of the P-ABB, an in-depth design of the location server is not being investigated at the current stage. The location server is stationary in the testbed. In the design of P-ABB, some other functions should be investigated in future that include, for example, an administrative interface for policy provisioning, and an inter-domain interface to operate a bilateral SLA negotiation.

In the testbed, those active nodes that do not function as P-ABB operate as S-ABBs. S-ABB may also work at edge or backbone router in the

networks. The S-ABB works as a helper in the decision-making process of the P-ABB, such as search filtering and policy conflict resolution. If an S-ABB is located at an ER, it sends COPS request to P-ABB and generates a separate LDAP request to LDAP server for fetching policy rule immediately upon receiving an SLA service request. Moreover, S-ABB can help in monitoring the network current status in terms of resource allocation, status of the network devices, update and auto-installation of a new module for network entity. Moreover, during a malfunctioning of the P-ABB, S-ABB can perform as a backup Policy Decision Point. With the support of dynamic module loading from the ASP, S-ABB can be upgraded to be the P-ABB upon the decision on relocation of P-ABB is made.

We have described the two main components in the ABB framework. Active networks play a significant role on facilitating agent mobility in our proposed architecture. Two important characteristics from active networks are noticeable as below.

1. Mobility of the P-ABB within the domain: With the network-wide class and module loading mechanisms provided by ASP, configuration and setup procedures of the BB and PEP can be done automatically with the minimum administrative intervention. This enables the P-ABB to relocate itself in the networks when congestion or malfunction of part of the network occurs.
2. Load sharing of policy decision making with network nodes: S-ABB at an active node is empowered with the ability of pre-computation on policy request. For example, as policy rules are being retrieved from the policy repository, active nodes can perform functions such as search filtering and policy conflict resolving on the traversing packet data. Refining of policy actions can be performed at S-ABB before the set of policy rules reaches the P-ABB, leaving a simpler decision of confirmation or denial of request to be done by P-ABB.

Load sharing allows distributive computations of the resource allocation decision, instead of locating more BBs in the domain or correspondingly reducing the size of a policy domain, we introduce active nodes in the networks that are able to reduce workloads at the decision point, i.e., the P-ABB. Moreover, a mobile P-ABB can evade the tragedy breakdown of centralized management service, and the networks can relocate a backup service as needed. Besides, P-ABB is able to detect the localized congestion situation, hence avoiding poor performance by moving to a less congested location. These properties can significantly improve scalability and availability of a policy control model. As a result, it simplifies the model structure and configuration procedures.

4. SYSTEM OPERATIONS IN ABB

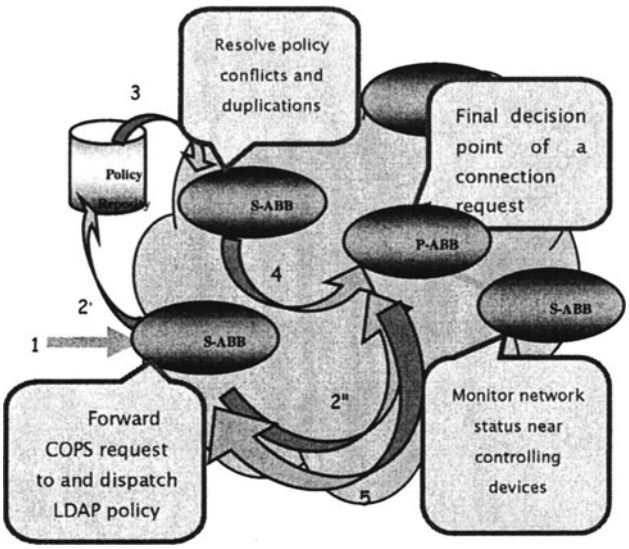


Figure 2. Operations of the Active Bandwidth Brokers Framework.

In ABB, both COPS and COPS-PR can be easily incorporated into the architecture. The design of ABB is transparent to both the outsourcing and provisioning models. At the current stage, a working network prototype[1] has been constructed with the initial focus on ABB scalability research. The initial work is to undergo the standard outsourcing model. A simple message flow sequence on service request and response procedures in the ABB framework is shown in Figure 2.

1. An SLA service request, sent from an end user or neighboring policy domain to this policy domain, arrives at an edge router.
2. The S-ABB at router receives this SLA request and generates two corresponding request messages. A COPS request message is sent to relay the service request to the P-ABB (marked as 2"). Another LDAP message is destined to the LDAP server for policy rule retrieval (marked as 2').
3. As soon as the LDAP request arrives at the LDAP server, the associated policy rules are retrieved and delivered to the P-ABB.

[1] There are 100 PCs running Linux operating systems. Among them, about 10 PCs function as active nodes. More time is needed to consolidate the testbed setup.

4. Other S-ABB agents along the path can do partial rule filtering and conflict resolution. The residual unresolved information is sent onwards to the P-ABB.

With the arrival of both the COPS request and the residual policy rules, the P-ABB makes the final policy decision, records this enforcement decision in P-ABB database, and sends it back to the edge router for policy enforcement. All packets that are related to policy protocols are encapsulated in ASP packets. The ASP header contains a data structure, AASpec [8], for controlling dynamic code loading. Typically, there are four different protocols used in the ABB model. Three of them are implemented as active applications, and they are the SLA, COPS and a modified Service Location Protocol (SLP) messages. Only LDAP messages are not encapsulated in active packets. The COPS messages in ABB framework are encapsulated in active packets. They are sent with UDP datagrams in the networks. The COPS' Request (REQ), Decision (DEC) and Delete Request State (DRQ) messages are implemented in the testbed.

5. EXPERIMENTAL RESULTS

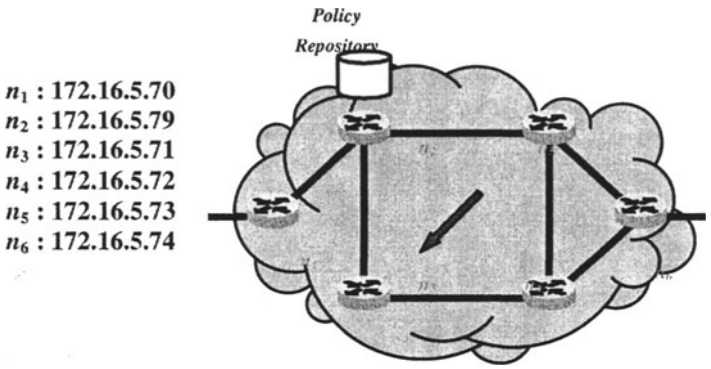


Figure 3. System Prototype for the ABB.

A network prototype has been constructed to verify the performance of ABB model. Figure 3 shows the network topology of the testbed at the time of experiments. There are six active nodes (n_1, n_2, ..., n_6). A network sniffer software package is used for collecting corresponding data on traffic flow in order to observe different network performance. All the active nodes are Pentium III PCs operating Linux kernel version 2.4.3. We deploy Active Signaling Protocol (ASP) v1.3 [9] as the execution environment (EE) for the active networks. These six active nodes function as edge routers receiving requests from a random request generator. The requests are uniformly

randomized in terms of their bandwidth requirements, the source address, the destination address, and the engaging time of the policies. The LDAP repository directory is stationary in the active network infrastructure, and it is hosted at the node n_2.

The following performance measures are defined to justify the proposed ABB design. They are:

1. Request Loss Rate, L_R: An end host sends a Service Level Agreement (SLA) request for establishing a new connection through an edge router. Because of the deficiency of the Transmission Control Protocol (TCP) in the installed ASP release, User Datagram Protocol (UDP) is used for passing COPS and LDAP messages. The PEP at the edge router may fail to receive the decision for policy enforcement if one of these messages is lost in the networks. Therefore, this metric is used to measure the failure rate of the connection requests.
2. Enforcement Delay, D_E: It is the time measured at the edge router with PEP entity from the time that it receives an SLA request to the time that the COPS decision message returns for enforcement.
3. P-ABB Decision Time, T_D: This delay measures the time required for making a policy decision at the P-ABB. It is the difference in time between the time that a set of policy rules arrives at the P-ABB and the time that its decision leaves the P-ABB.

5.1 Stationary and Mobile P-ABB

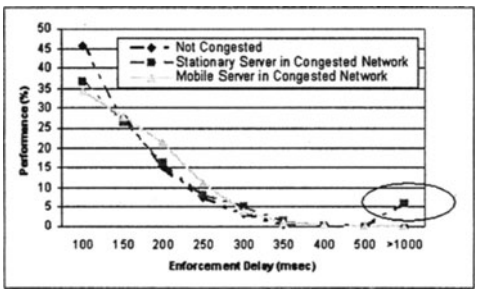
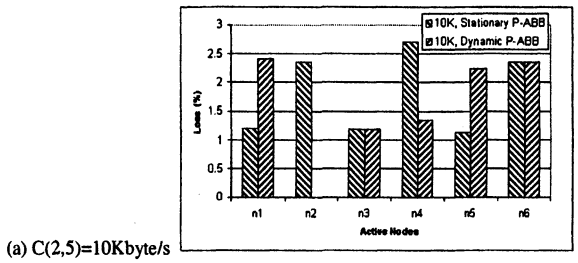
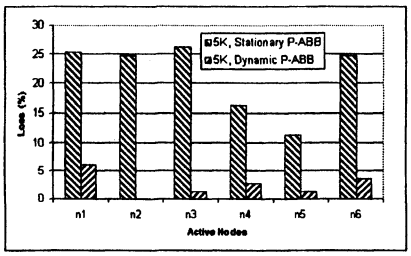


Figure 4. Stationary/Mobile P-ABB.

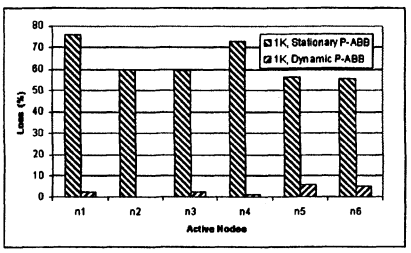
In these experiments, we examined the difference that would be made with the mobility of the P-ABB. From the system prototype as shown in Figure 3, the P-ABB was located at the node n_5. The capacity of the link between nodes n_2 and n_5, $C_{(2,5)}$, could be modified with the Class-Based Queueing (CBQ) in Linux.



(a) C(2,5)=10Kbyte/s



(b) C(2,5)=5Kbyte/s



(c) C(2,5)=1Kbyte/s

Figure 5 . Request Loss Rate under Different Capacities of the Bottleneck Link.

In the first set of tests, a given amount of traffic will be randomly generated. We studied the Enforcement Delay, D_E, through three different scenarios. The performance in terms of ratio of all requests would become noticeable when the bandwidth of link (2,5) went down significantly from 100Mbps to 1Kbps. The D_E was the average measured result over one thousand requests. Three sets of measurements were taken. For the first set, the experiment is performed in a non-congested environment. For the second set of tests, the link (2,5) became congested, and the mobility of P-ABB was disabled. The testing environment was similar to the PBM framework from IETF. For the third set of tests, P-ABB would move and it went to the node n_3 during the testing.

The measured three sets of results were plotted as shown in Figure 4. We noticed that the improvement of the mobile agent was significant. There was more than 5% of traffic that had suffered D_E of more than 1sec in the testing of congested networks with stationary P-ABB. With the mobility enabled, no significant delay, D_E, was noticeable. In Figure 4, they were shifted downward to the range around 200msec. The next set of tests examines the effect of mobility of the P-ABB on the Request Loss Rate, L_R. In comparing the mobility effect on the system performance, the measured results are plotted in Figure 5(a), (b) and (c) for $C_{(2,5)}$ = 10KBps, 5KBps and 1KBps, respectively. The improvements are noticeable. From Figure 5(c), the bottleneck link in networks had only 1KBytes/sec rate, the loss rate was higher than 50% for stationary P-ABB, whereas the loss rate for mobile P-ABB were less than 5% for all nodes. From these experiments, the mobility of P-ABB improves the system significantly in both L_R and D_E. Hence, the proposed ABB improves PBM on both the scalability and reliability issues.

5.2 Effects of S-ABBs

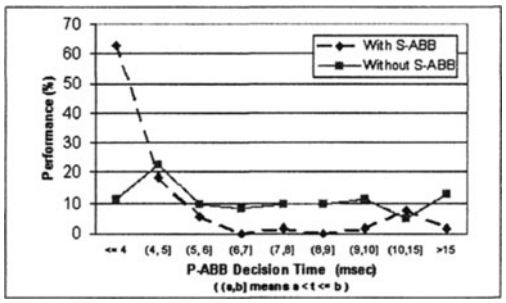


Figure 6. Effect on Decision Time with/without S-ABB.

In this subsection, we examine the effect of S-ABB in the proposed ABB framework. For the study of the P-ABB Decision Time, T_D, five hundred random SLA requests are made and sent to each of the six PEPs. Moreover, the P-ABB was made stationary in this set of experiments. Initially, the functions for all S-ABB agents were disabled. The measured mean time of T_D was 8.48msec. For the next set of tests, we enabled all the functionalities of S-ABBs in the testbed, the measured mean time of T_D was 7.63msec. The two measured results were plotted in Figure 6. The impact of S-ABB was significant after its functionalities were set in place, because about 60% of the decision time at P-ABB was made less than 4msec.

6. CONCLUDING REMARKS

A novel design for the Policy-Based Management, the Active Bandwidth Broker, has been proposed. There are two operating entities in the framework: the primary ABB (P-ABB) and secondary ABB (S-ABB). P-ABB is the mobile agent in the architecture, which is responsible for making the final decision on policy enforcement upon each connection service request. More importantly, it can relocate itself to another active node through active network technology upon detecting different congested conditions. The active node with moved P-ABB converts itself to operate as an S-ABB. Currently, P-ABB may be considered as a single point of failure in the architecture. However, the location server checks if the P-ABB is functioning, and it may be trigger another S-ABB to function as an P-ABB if the original one is lost. However, a more sophisticated design will be introduced in the future to support multiple P-ABBs, and thereby increase system reliability. The conditions for creating and destructing P-ABBs should be considered. As well as the communication mechanisms among all P-ABBs should be designed to synchronize network information. The results of a working experimental prototype demonstrate the feasibility of the ABB model. To obtain a global view of the network performance at one particular instant is difficult because of network processing and propagation delays, but it should be one of our future goals to locate the best active node across the domain to host the P-ABBs.

Reference:

[1] R. Yavatkar, D. Pendarakis, R. Guerin, *A Framework for Policy Based Admission Control*, IETF RFC 2753, January 2000.

[2] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Raja, A. Sastry, *The COPS (Common Open Policy Service) Protocol*, IETF RFC 2748, January 2000.

[3] K.H. Chan, D. Durham, S. Gai, S. Herzog, K. McLoghrie, F. Reichmeyer, J. Seligson, A. Smith, R. Yavatkar, *COPS Usage for Policy Provisioning (COPS-PR)*, IETF RFC 3084, March 2001.

[4] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Raja, A. Sastry, *COPS usage for RSVP*, IETF RFC 2749, January 2000.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An Architecture for Differentiated Services*, IETF RFC 2475, December 1998.

[6] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, IETF RFC 2205, September 1997.

[7] M. Wahl, S. Kille, T. Howes, *Lightweight Directory Access Protocol (v3)*, IETF RFC 2251, December 1997.

[8] D.L. Tennenhouse, D.J. Wetherall, "Towards an Active Network Architecture," in *Multimedia Computing and Networking*, 1996.

[9] Active Reservation Protocol: http://www.isi.edu/active-signal/ARP.