Policy-based Traffic Management in Home Area Network – An Elementary Testbed Model

Annie Ibrahim Rana
Telecommunications Software and Systems Group
Waterford Institute of Technology
Waterford, Ireland.
arana@tssg.org

Micheal O Foghlu
Telecommunications Software and Systems Group
Waterford Institute of Technology
Waterford, Ireland.
mofoghlu@tssg.org

ABSTRACT
Traffic management in home area network (HAN) is different from the traditional traffic management in access and core networks. Traditionally network traffic works in best effort fashion and the HAN services are usually accommodated on the basis of first-in first-out rule. However quality can deteriorate when high number of users is connected to the HAN. Moreover the bursty traffic can also impact the quality by chocking the network traffic and blocking the network resources for all other traffic. Traffic management rules can be employed in HAN to prioritise different types of traffic according to user requirements. Dynamic configuration of network resources and services is multifaceted process, which requires many skills and knowledge. Policy-based Traffic Management (PBTM) can play a significant role in managing home networks and configuring the services dynamically according to HAN user requirements. This paper presents a testbed model for HAN to simplify traffic management process based on the principles of policy-based network management.

Keywords
Network Management, Policy, Policy-based network management, network traffic prioritization. Policy-based traffic management, autonomic network.

1. INTRODUCTION
In a traditional home area network (HAN), there can be several types of network traffic e.g. VoIP, Audio & Video on demand, Web and many more. Usually the HAN traffic works in best effort fashion meaning QoS is not guaranteed. HAN traffic quality can deteriorate due to bursty traffic, which usually deprives off all other traffic from utilising network resources due its greedy nature; sometimes applications, devices and users connected to the HAN require network resources more than a network capability, in such situations no service gets satisfactory share in network resources. This leads the network into a state of congestion, which sometimes chocks network traffic flow and results in poor quality of network services. Mostly the solution to resolve congestion issue is considered in getting more bandwidth for the network but logically it alleviates the issue temporarily but doesn't provide any long lasting remedy for a healthy networking capability.

FIT'09, December 16–18, 2009, CIIT, Abbottabad, Pakistan.

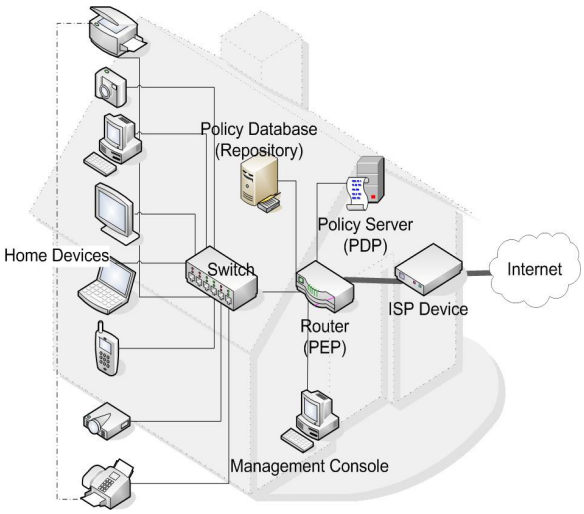

Figure 1: PBTM in HAN.

The most important fact, which is usually ignored in network management, is the HAN user requirement. A user, who is having video conference online and downloading some stuff simultaneously, may want to prioritise video conference over downloading traffic to have better video quality. Similarly two users accessing the HAN, administrator may want to give higher priority to the traffic of one user over the other. A traffic management scheme (e.g. prioritisation), which is not a new concept, can be employed in such situations to manipulate the traffic priority statistically and to treat the traffic accordingly based on the classification ranks. This concept is more or less related to differentiated services (DiffServ) but in our case it is restricted to HAN domain that means the traffic may not be treated in similar fashion out side home domain. However if this concept is extended to other cross domains e.g. internet service provider, then QoS can be further improved.

In this paper we present a testbed model for HAN traffic management based on the principles of policy-based network management. Figure 1 shows the role of policy-based network management in HAN. The residential gateway device or the router

is policy execution point to manage different types of traffic according to HAN users.

This paper is structured as follows: In first part we briefly summarise important concepts related to QoS and traffic management, and policy-based traffic management; in the second part we discuss related and present the testbed model and lastly conclude the research work with future work directions.

2. QUALITY OF SERVICE MANAGEMENT

Quality of service management (QoSM) is network management technique to configure and maintain services and network resources to achieve network quality requirements. QoSM is usually attained through controlling the traffic and reserving the resources. It uses priority rules to provide a certain level of service based on the priority of different classes of users, applications and traffic flows. For guaranteed services it allocates recourses to particular traffic class. QoS is a collective measure of the level of network service provided to a user, which can be characterized by many performance parameters of a network:

1. Timeliness characteristics
2. Capacity characteristics
3. Error-related characteristics
4. Reliability characteristics
5. Security characteristics
6. Cost characteristics etc….

However most commonly used parameter are three:

1. Delay – It refers to a lapse of communication data in terms of time between two points resulting from queuing, processing and congestion.
2. Jitter – It refers to variations in a data communication resulting from fluctuations in the flow, also called as distortion
3. Loss – It refers to loss of the transmitted data packet usually resulting from data congestion at some point along the network path.

QoSM helps to setup and evaluate QoS goals (policies); QoS policies are transformed into configurations, which act as networking rules. A QoSM methodology entails baselining the network deploying relevant QoS techniques and evaluating QoS results.

2.1 QoS Levels

QoS level, also referred as service level, is network QoS capability to deliver service needed by network traffic. QoS can be graded into three basic levels [10]:

- Best-effort service level – This is also known as lack of QoS, best-effort service is basic connectivity with no guarantees.
- Differentiated service level – This also known as soft QoS. Different traffic are classified and treated according to their classification. This is a statistical preference, not a hard and fast guarantee.
- Guaranteed service level – This also known as also called hard QoS. It reserves network resources for specific traffic.

2.2 QoS Types

There are two types of QoS [11]: provisioned QoS and signalled QoS. Provisioned QoS is statically achieved by configuring network resources for the flow of different types of traffic. Most of QoS approaches are static using priority queues, data flow control and packet marking etc. In signalled QoS, which is also referred as dynamic QoS, the IP packets contain signalling information describing the specific QoS necessary for the application to function. The Resource Reservation Protocol (RSVP) protocol is mostly used for signalled QoS.

QoS manage traffic in two ways [11]: per-flow QoS, and per-aggregate QoS. Flow is unidirectional stream of data, which receives individual treatment in per-flow QoS. In per-aggregate QoS, two or more unidirectional data streams are put under some classification based on some traffic characteristics e.g. all packets using tcp protocol, and the class of different flows receives individual QoS treatment. Provisioned QoS used aggregated QoS traffic management technique and signalled uses per-flow technique. Both QoS techniques can be used in other ways with per-flow and per-aggregate but in that case they may not make much sense.

2.3 QoS Architecture

The basic architecture introduces the four fundamental elements QoS traffic management:

- Traffic identification scheme
- Traffic marking scheme
- Traffic filtering scheme
- Traffic queuing scheme

Traffic identification is usually based on the information available in traffic packets and the QoS implementation technique .e.g. source & destination IP addresses, ports, protocols etc. To provide preferential QoS treatment to a type of traffic, it must be identified first. Traffic marking is not compulsory because traffic can be filtered for QoS treatment even if it is not marked but it really depends on how QoS is implemented. Generally TOS bits in IP packets can be marked for different types of QoS treatments. Traffic identification and marking together called as traffic classification. When the packet is identified but not marked, classification is said to be on a per-hop basis. This is when the classification pertains only to the device that it is on, not passed to the next router. QoS implementation technique depends on the user QoS requirements. User can go for priority-based queues, class-based queues etc.

2.4 Traffic Queuing Mechanisms

QoS techniques use different queuing mechanisms and some of them are very standardized and widely used. We briefly discuss the queuing mechanisms we used in our testbed model:

2.4.1 Priority Queuing (PQ)

It is the basis for a class of queue scheduling that is designed to provide a relatively simple method of supporting differentiated service classes. In PQ, packets are first classified by the system and then placed into different priority queues. Packets are scheduled from the head of a given queue only if all queues of higher priority are empty. This is also referred as per-aggregate queue.

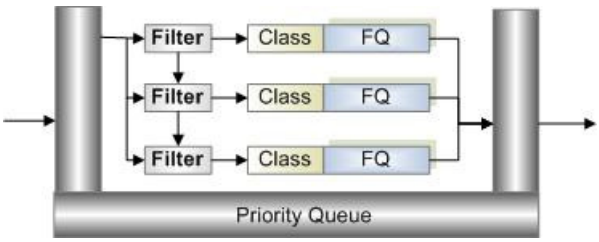

Figure 2: Packet Queue Manager

2.4.2 Fair Queuing (FQ)

It is the foundation for a class of queue scheduling that is designed to ensure that each flow has fair access to network resources and to prevent a bursty flow from consuming more than its fair share of output port bandwidth. In FQ, packets are first classified into flows by the system and then assigned to a queue that is specifically dedicated to that flow. Queues are then serviced one packet at a time in round robin order. Empty queues are skipped. FQ is also referred to as per flow or flow based queuing.

3. POLICY-BASED TRAFFIC MANAGEMENT

Policy-based traffic management (PBTM), a sub-domain of policy-based network management (PBNM), is management paradigm in networking that separates administration operations from other basic network operations. It provides a flexible and robust mechanism to allocate network resources and services like bandwidth allocation, quality of service, access rights, traffic prioritization and security to different network elements. It results in increasing quality of work, efficiency, adaptability, coherent network behaviour, flexibility and reduced maintenance cost regarding to network management [3, 4]. It is often part of a wider autonomic networking approach (i.e. self-governing) (c.f. [1]) that aspires to reduce the human intervention, reduce cost and reduce errors.

3.1 Policy Definition

There is no standard way of defining policy but there are some definitions put forward by academic researchers. According to [6], policy is predetermined action statement for such action patterns that are repeated by entities involved in a network under certain systems conditions when they are met. The paper [7] defines policy as a goal or course of action to guide present and future network decisions. More concisely, policy is set of rules to administer, manage and control the access to network resources and services.

There are mainly two types of network operations: Core network operations, management operations. Network management can be further broken into three major types of management tasks: Network QoS Management, Network Security Management, and Network Configuration Management. QoS and security, both requires configuration management and are dependent on it. However network policies can be classified generally into the following six broad categories [5]:

1. Performance Management Policies
2. Security/Access Control Policies
3. Quality of Service Policies
4. Administrative/Configuration Management Policies
5. Fault Management Policies
6. Customized/Event Condition Action Policies

In this report we are focusing on QoS management policies for home area network. A policy gives abstraction to control network resources/elements. By the using policy-based QoS management, network resources can be used efficiently. We would discuss the benefits of this approach in later sections.

3.2 PBTM Work Model

Policies are created, modified and stored in repository through policy management service using policy management console. Policies are stored in repository. Stored policies are retrieved by policy decision point server and enforced at policy enforcement points, the network elements (router, bridges, servers, desktop etc.) Figure 2 shows very simple PBNM work model.

High level/Abstract policies are translated into specification level policies. Policy translation can be done by using policy specification language, rule based approach or formal logic based approaches.

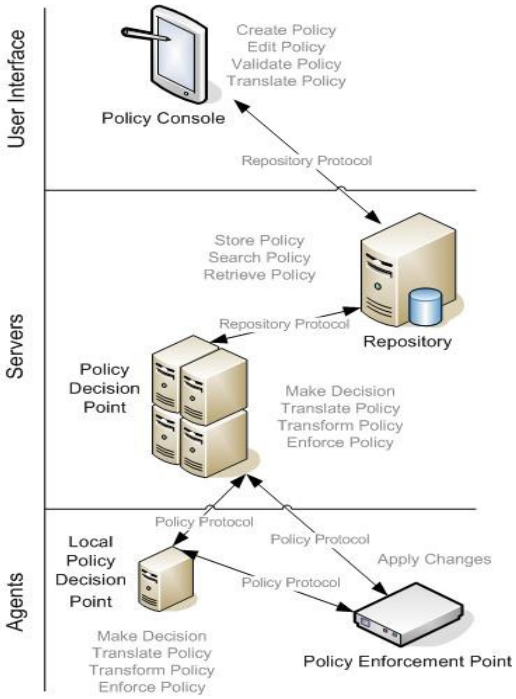

Figure 3: PBTM work model.

Specification level policies are further transformed into low level policies / configurations, which are applied to network devices/agents. When any triggering event happens, new policy decisions are made and applied to network automatically as shown in figure 2.

4. RELATED WORK

Extensive significant work has been done to manage QoS requirements in access and core networks using PBNM. Different architectures are proposed in [2] for the control plane of a software router that integrates signalling protocols and control mechanisms for QoS and in [9] using PBNM. The paper [2] claims that the use of proposed architecture can meet the end-to-end QoS requirements for most Internet applications if applied on

the access network routers. However, it is worth discussing issue that how the end-to-end QoS requirements can be met with out ensuring QoS at the edge devices in residential networks.

Traditional PBNM systems focus on the management of core networks and the internet in the broader sense. The access and the core networks use policies to meet Service Level Agreements (SLAs) for different service users. However the concept of end-to-end QoS in the big picture would remain in a status-quo if QoS is not ensured at the edge networks (Home Networks, Office Networks etc.). PBNM can play a significant role in managing home networks focusing on users' requirements. Lets suppose if we have one intelligent gateway device to control all outgoing and incoming traffic, which can be configured according to user requirements through a policy manager; it would make HAN users' life much easier. The paper [3] proposes similar solution but it focuses more on Intelligent Control Centre (ICC) to connect all other networks with in HAN e.g. Power Line Network, PC Network, Wireless Network, Home Automation Network, and Home Gateway. On the other hand what we proposed and implemented is an intelligent gateway machine, which configure itself according to changing user requirements.

5. TESTBED CONFIGURATIONS

We have simulated HAN in our research lab, and our research experiments testbed uses the settings and configurations as discussed below.

5.1 Equipment and Applications

We used a Linux machine with Ubuntu Linux distribution as a gateway (software router) for HAN. We used traffic control (TC) for setting up filters and queues using shell scripts on the router. IPTables package is used for defining NATing, routing and prioritisation rules using shell scripts, and TCPDump for packet analysis using perl script.

5.2 Networking Configurations

The router has two interface cards:

1. Eth0: 10/100 Ethernet for LAN connectivity
2. Eth2: 10/100 Ethernet for Internet connectivity

Eth2 is automatically configured with Dynamic Host Configuration Protocol (DHCP) server and Eth0 is manually configured in interface configuration file.

5.3 NAT and Routing Configurations

We have locked all the services on the router so that only LAN traffic can access internet and traffic generated by router is blocked. We have allowed Secure Shell (ssh) traffic to from WAN to access router machine. We allowed Web, Voip and FTP traffic on LAN. To make our network secure we blocked all other incoming traffic

5.4 QoS Configurations

We created priority queue using TC application. Based on the user requirements (as per test scenario description), traffic classes are created for voip, http and other (ftp). For each class of traffic, stochastic fairness queue (SFQ) is attached to manage the packets. Policy rules for traffic filtration and marking are devised from the high level user requirements/goals.

6. TESTBED ARCHITECTURE

Due to massive increase in HAN traffic load, high usage of web applications and lack of knowledge of HAN users about networking management, have made the scope of policy-based traffic management in HAN more important than ever before. Web applications in the HAN are mostly multimedia-intensive with different quality and security requirements; especially audio and video applications traffic is more sensitive to delays and packet loss. When multiple applications are running on a network then managing network resources -e.g. bandwidth, is non-trivial task, especially deciding about what resource goes to which of the applications.

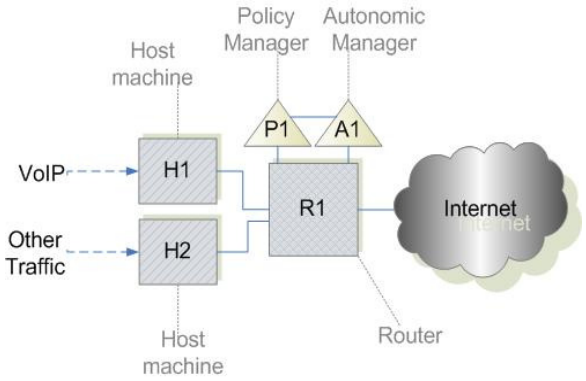

Figure 4: Test Scenario Architecture

The simplest solution to this problem lies in traffic prioritisation; residential gateway (router) can be used to manage network traffic requirements. Traffic management in HAN should always be applied on shared media access, which has high risk of becoming a bottleneck at the time of network congestion. Point-to-point media access doesn't require traffic management techniques. Traffic management through prioritisation is optimal technique for differentiated traffic, media, applications and users. Though prioritisation technique doesn't provide guaranteed service contrary to parameterised QoS, but it fits better in HAN due to flexibility in configuring the priority for different network traffic.

In our test scenario we used Linux machine as a gateway device, which differentiate different types of traffic based on their IP addresses and protocols, and then packets are marked for further QoS treatment. Figure 4 shows the architecture of our test scenario. Queues are created to manage high and low priority traffic. We used three-priority queues to manage the traffic. On each queue a class filter is added to classify the traffic based on the TOS bits markings in packet IP header. Once traffic is identified and classified, it is queued into the respective priority queue and then scheduler automatically manages the queues according to their priority ranks. We used to Traffic Control (TC) Linux application to create filters and queues. In figure 3 R1 represents Ubuntu Linux machine. H1 is the host machine which is connected to R1 through Ethernet 10/100. H1 runs different types of traffic -e.g. voip, ftp, http etc. According to policy rules, different traffic packets' TOS bits are marked in their IP headers. We categorized the traffic into three priority groups: all Session Initiation Protocol (SIP) and Real Time Protocol (RTP) traffic with highest priority, all http and https with medium priority and rest with lowest priority.

6.1 Policy Builder

Policy builder provides a simple interface to define high level traffic management requirements. At this stage policy builder only allows prioritising three classes of traffic. The high level requirements are then transformed into policy rules and configuration scripts are generated accordingly. We haven't used any formal policy language for specification. Currently we are using IPtables INFPUT (for incoming traffic) and FORWARD (routing between LAN and WAN interfaces) chains rules defining NATing and routing policy rules. For traffic prioritisation, we have used Traffic Control (TC) application for defining the priority queues over the WAN interface; we are using IPtables managle table for defining packet marking and forwarding rules.

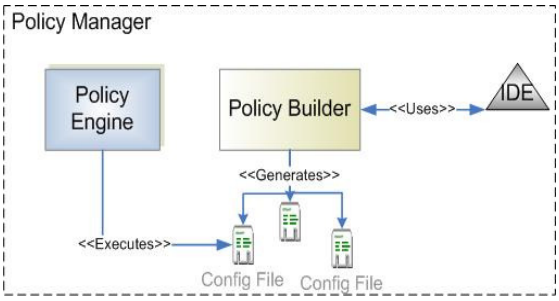

Figure 5: Policy Manager

In future more advance features would be introduced in policy builder and formal policy specification language would be used. Policy builder is component of policy manager as shown in figures 3 and 5.

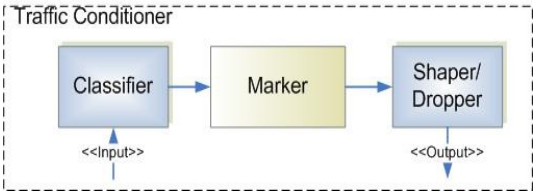

Figure 6: Traffic Conditioner

6.2 Policy Engine

Policy engine is also the part of policy manager. The policy engine executes the scripts generated by policy builder. Policy engine also monitor the changes in policy rules, if there is any change in configuration scripts, those changes are directly applied to the router. At this stage, policy engine is a simple crontab, which executes automatically for first time.

Figure 5 shows the working model of policy builder and policy engine. Policy builder uses an IDE to define traffic prioritisation goals and then those goals are transformed into policy rules. At this stage policy IDE, builder and engine all provide very limited features.

6.3 Traffic Conditioner

We used simplest method of marking the packets using type of service (TOS) bits in IP header. The traffic conditioner is deployed at HAN gateway, the router. The conditioner consists of a multi-field (MF) classifier and a marker; in our case we haven't used meter as a part of conditioner that normally takes a measure of the incoming traffic throughputs that are previously classified by the classifier. Packets are marked by setting TOS bits according to policy rules and then they passed through the shaper/dropper for further processing and then pushed to their respective queue, if not dropped by the dropper. Figure 6 shows the architecture of traffic conditioner.

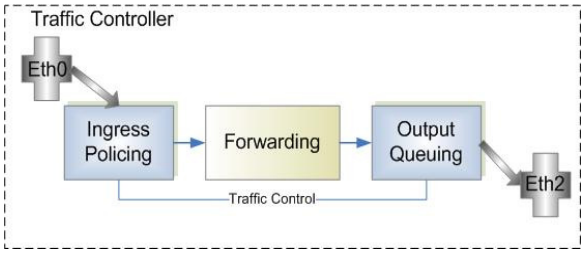

Figure 7: Traffic Controller

6.4 Traffic Controller

We used two network interface cards on a Linux machine (router), which are driven by network drivers to control the hardware. The network drivers act as exchange mechanism of packets between the Linux and the physical network. We used TC application to control and IPtables package to control and manage the traffic. Figure 7 shows traffic controller architecture.

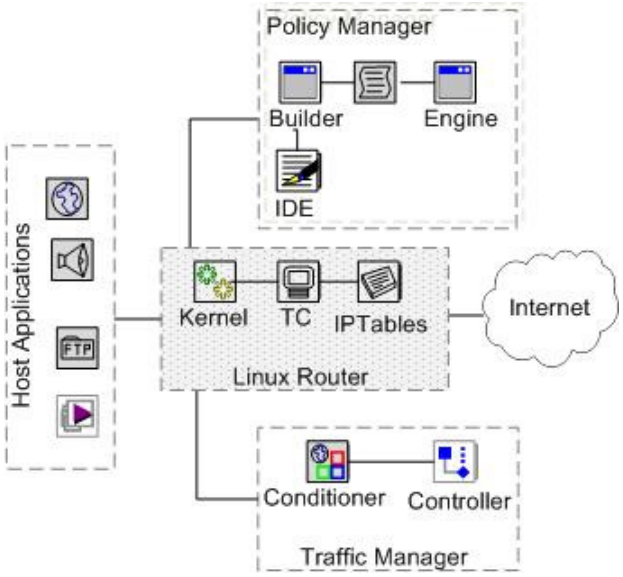

Figure 8: Testbed Model

6.5 Queue Manager

To keep traffic management simple, we used priority queue discipline that contains an arbitrary number of classes of different priority. We created three classes and for each class there is 1 queue based on stochastic fairness queuing (SFQ) protocol. Figure 2 shows the architecture of priority queue used in our testbed.

6.6 Queues Analyzer

Queues analyzer runs as a backend process and displays the status of each traffic queue created on the router's WAN interface. It shows number of packets queued and dropped at each queue.

7. CONCLUSION

Policy-based traffic management provides a flexible and robust mechanism to allocate network resources, quality of service, and traffic prioritization to different network traffic. It is a better approach to meet HAN user requirements to manage the network traffic. The best advantage of using PBTM is that policies can be changed at run time without affecting underneath working model. It means traffic management policies can be changed dynamically and it is the very basic challenge in managing HAN traffic because user requirements can change time to time.

In this paper we have presented a testbed model for policy-based traffic management in HAN, and we have implemented this testbed for our future research work. Figure 8 shows testbed model. This model can be used in HAN to address dynamic traffic management requirements and the related issues –e.g. prioritisation, bandwidth allocation, and traffic shaping and dropping.

8. FUTURE WORK

This is the initial work for our project; this was to setup a test environment for our future research work. However, this testbed would be further refined and built to the next level for future research requirements. The interesting aspects of the testbed that we would be looking at in HAN traffic management are:

1. Use of policy continuum [8] and formal policy specification.
2. Use of autonomic policy refinement techniques [1].
3. Building traffic management tool for HAN users.

9. ACKNOWLEDGEMENT

The authors wish to acknowledge the support of the SFI SRC FAME (Ref: 08/SRC/I1403) award that contributed financially to the work that is reported in this article.

10. REFERENCE

[1] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. OFoghlu, W. Donnelly, and J. Strassner. Towards autonomic management of communications networks. Communications Magazine, IEEE Publications, 45(10):112–121, October 2007.

[2] J. Maniyeri, Z. Zhang, R. Pillai, and P. Braun, 2003. A Linux Based Software Router Supporting QoS, Policy Based Control and Mobility. In Proceedings of the Eighth IEEE international Symposium on Computers & Communications (June 30 - July 03, 2003). ISCC. IEEE Computer Society, Washington, DC, 101.

[3] G. Liu, S. Zhou, X. Zhou, and X. Huang, 2006. QoS Management in Home Network. In Proceedings of the international Conference on Computational intelligence for Modelling Control and Automation and international Conference on intelligent Agents Web Technologies & international Commerce (November 28 - December 01, 2006). CIMCA. IEEE Computer Society, Washington, DC, 203.

[4] R. Boutaba and I. Aib. Policy-based management: A historical perspective. ACM Journal of Network and Systems Management, 15(4):447–480, December 2007.

[5] S. Boros. Policy-based network management with snmp. In Proceedings of EUNICE, pages 13–15. University of Twente, Netherlands, September 2000.

[6] J. Saperia. IETF Wrangles over Policy Definitions. Network Computing, IETF Policy Framework Working Group, 2002.

[7] A. Westerinen. Terminology for policy based management. ACM IETF RFC 3198, 2001.

[8] S. Davy, B. Jennings, and J. Strassner. The policy continuum - a formal model. In Proceedings of the Second IEEE International Workshop on Modelling Autonomic Communications Environments, pages 65–79. MACE, March 2007.

[9] A. Ponnappan, L. Yang, R. Pillai, and P. Braun, 2002. A Policy Based QoS Management System for the IntServ/DiffServ Based Internet. In Proceedings of the 3rd international Workshop on Policies For Distributed Systems and Networks (Policy'02) (June 05 - 07, 2002). POLICY. IEEE Computer Society, Washington, DC, 159.

[10] Cisco, Quality of Service (QoS), CiscoPress, Cisco, 2006.

[11] Hewlett-Packard. A Primer on Policy Based Network Management. Open View Network Management Division, Hewlett-Packard Co., 19999.