Evaluation of a Policy-Based QoS Management Architecture over an IPv6 DiffServ testbed

Alex Vallejo, Agustín Zaballos, Jaume Abella, Guillermo Villegas and Josep M. Selga, *member, IEEE*
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull (URL)
Barcelona, Spain
{avallejo, zaballos, jaumea, gvillegas, jmselga}@salle.url.edu

Abstract— **In this paper we present the implementation of a bandwidth broker architecture for the centralized management of the QoS in native IPv6 DiffServ networks in a multi-domain environment. The bandwidth broker has been programmed in JAVA and uses COPS-PR as an intra-domain communication protocol and SIBBSv6 as an inter-domain communication protocol. The latter protocol is also presented. SIBBSv6 is a variation of the SIBBS inter-domain communication protocol with some new features such as the IPv6 support. The evaluation of the proposed architecture has been tested by implementing a real IPv6 DiffServ testbed composed of three domains. Each of these domains is managed by the developed bandwidth broker named BBv6. The results of these tests and the corresponding conclusions are also presented.**

Keywords- IPv6; QoS; Differentiated-Services; Policy Based Networking Management; COPS-PR; SIBBS

I. Introduction

Two of the main concerns in Mobile IPv6 technology are QoS and IPv6 in which they are needed in both core and access networks. In fact, they are also a common feature in the Next Generation Networks (NGNs).

IPv6 is currently supported by the main software and hardware applications, at least at the basic stage, and is present in the main worldwide networks. Moreover, as a result of the work carried out by the USAGI project [1], which has merged its work into the official Linux kernel, IPv6 stack in Linux OS is now fully compliant with advanced IPv6 conformance and interoperability tests [2][3]. Thus, real IPv6 testbeds with Linux OS can be implemented with performance guarantees.

The QoS management for the new IP technologies in NGNs has undergone important advances with the introduction of the Policy-based network management (PBNM) for resource allocation. NGNs require automatic provisioning of QoS and PBNM simplifies the definition and deployment of network policies through centralized management frameworks. Therefore, the centralized management of networks with QoS and/or Traffic Engineering has become a fundamental issue in recent research. Some proposals are the Bandwidth Brokers (BB) [4] for DiffServ networks or the Path Computation Elements (PCE) [5] for MPLS-TE.

The BBs are devices capable of automatically managing the QoS using a centralized architecture within a DiffServ domain which have been proposed for the control and management of QoS provisioning to reduce the complexity of the control plane. They are basically resource controllers which manage the limited amount of resources specified by the client contracts or the Service Level Agreements (SLAs) of a DiffServ domain and make the service allocation decisions from those resources to be applied to the nodes of that domain. The BBs allow the administrator to configure network policies with a high level language and a friendly interface and save these policies in a structured way for the different nodes involved. They are then configured within these nodes using specialized policy-based transmission protocols, making them transparent to the administrator. These architectures are able to manage DiffServ networks using an intra-domain communication protocol for policy delivery and are able to interact with other BBs in other DiffServ domains by using an inter-domain communication protocol.

This paper expands on a previous study [6] in which the authors presented a bandwidth broker architecture for the centralized management of a single IPv6 DiffServ domain through an intra-domain communication protocol. This paper presents the implementation of a bandwidth broker architecture for the centralized management of the QoS in native IPv6 networks in a multi-domain environment and therefore, the implementation of the inter-domain signaling among BB peers in different IPv6 DiffServ domains. The performance has been analyzed in a real testbed with three DiffServ domains which have IPv6 networks.

The paper will proceed as follows. In section II we provide the background for the network architecture required to reach the objectives. In section III, we describe the BBv6 and the new inter-domain module. In section IV we describe the testing environment, the policies used and the evaluation results of the tests. Finally, we make our concluding remarks and comment on future work in the last section.

II. Network architecture

DiffServ networks were introduced to solve the implementation and deployment difficulties of IntServ networks. The Differentiated Services Working Group (DiffServ WG) [7] defined an architecture based on pushing complexity to the edges of the network and keeping classification and packet handling functions in the core network as simple as possible [8]. Moreover, DiffServ (DS) offers more scalability of QoS provisioning than the IntServ networks because "the amount of state information is proportional to the number of classes rather than the number of flows" [9].

In these kinds of networks a customer may be a user organization (source domain) or another DS domain (upstream domain). Both must have a service contract, or SLA, with its ISP to receive differentiated services where the service classes supported and the amount of traffic allowed in each class will be specified through the Service Level Specifications (SLSs) [10]. Therefore, these SLSs are the set of parameters which define the services offered to the traffic flows by the DiffServ domains and are composed of Traffic Conditioning Specifications (TCS) [10] which specify the set of classification rules and the traffic profiles.

The flows are classified by the MultiField (MF) classifier, and then metered, policed, marked and shaped at the edge nodes of a DiffServ domain. The core nodes handle packets according to Per Hop Behaviors (PHBs) [8] which are selected on the basis of the Behavior Aggregate (BA) classifier which selects packets based exclusively on the DS field contents. The DiffServ codepoint (DSCP) [11] in the DS Field maps the class of service in every IP packet header, IPv4 and IPv6. Therefore, the PHB is the forwarding treatment applied to a collection of packets with the same DSCP which cross a link in a particular direction at a DiffServ-compliant node. When a packet ingresses into a DS domain from another DS domain, its DSCP may be re-marked according to the SLA between the two domains.

A. *Intra-domain management*

The IETF's Policy Framework Working Group (Policy WG) [12] proposed anarchitecture for the policy-based network management providing a framework to enable centralized control of a domain which is independent of the devices and protocol that form it. The architecture proposed is composed of a Management Console, a Policy Repository, a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP). An intra-domain communication protocol is needed for the transmission of policy decisions from the PDP to the PEP within a domain. This was standardized by the Resource Allocation Protocol Working Group (RAP WG) [13] on the development of COPS (Common Open Policy Service) [14].

COPS is a query/response protocol (stateful) that uses TCP and supports two common models for policy control: Outsourcing and Configuration. The latter, COPS-PR (Common Open Policy Service for Policy Provisioning) [15], is the COPS evolution to cover the DiffServ model needs. In this model the PDP may proactively provision the PEP and both have a virtual container called PIB (Policy Information Base) where the policies are stored. This PIB has a tree structure formed by PRovisioning Classes (PRCs) which contain PRovisioning Instances (PRIs) [16]. Once the PEP has been initiated, and whenever there are updates, the appropriate policies are sent out by the PDP (Fig. 1). This way the PDP keeps the two PIBS synchronized.

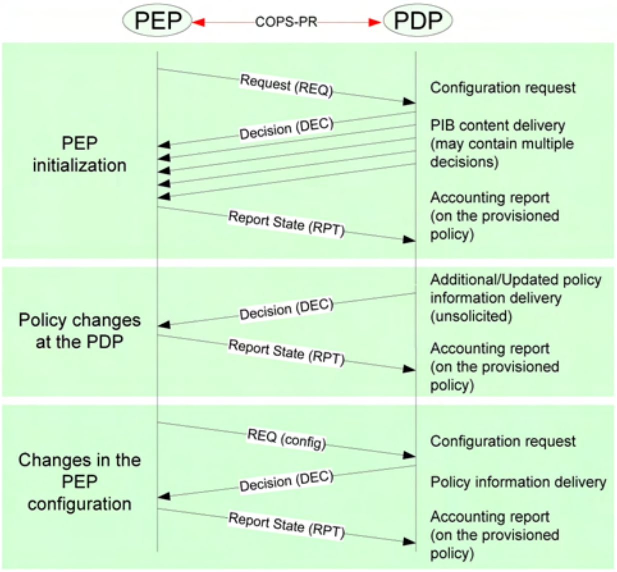
PEP
COPS-PR
PDP
PEP initialization
Request (REQ)
Configuration request
Decision (DEC)
PIB content delivery (may contain multiple decisions)
Report State (RPT)
Accounting report (on the provisioned policy)
Policy changes at the PDP
Decision (DEC)
Additional/Updated policy information delivery (unsolicited)
Report State (RPT)
Accounting report (on the provisioned policy)
Changes in the PEP configuration
REQ (config)
Configuration request
Decision (DEC)
Policy information delivery
Report State (RPT)
Accounting report (on the provisioned policy)

Figure 1. COPS-PR signaling between the PDP and the PEP

There are other intra-domain protocols for network policy delivery, such as the one developed by the Configuration Management with SNMP Working Group (SNMPConf WG) [17], in charge of mapping the Policy WG framework to SNMP, defining the Policy Based Management MIB [18]. However, as stated in [6], COPS-PR/PIB has some advantages over SNMP/MIB.

B. *Inter-domain management*

The first organism to define the basic architecture of a BB was the Internet2 QBone Bandwidth Broker Advisory Council (I2-QBBAC) in the year 2000 [19] which stated that a BB must be composed of an inter-domain communication interface, an intra-domain communication interface, a database that contains the network topology and of an external or embedded QoS/policy management service which should be based on SLAs and Resource Allocation Requests (RARs). Moreover, they developed SIBBS (Simple Inter-domain Bandwidth Broker Signaling), the first inter-domain signaling protocol for this kind of architectures.

Some proposals have been made for inter-domain communication protocols. RSVP [20], BGRP [21][22], DSNP [23], DARIS [24], COPS-SLS [25], COPS-DRA [26] and SIBBS are the more noteworthy examples. As scalability and aggregation are not a factor and we are working in centralized environments with SLA/SLS, the most suitable election would be COPS-SLS and SIBBS, even though neither of them are standardized.

SIBBS is a very simple TCP based protocol to be used between BBs. A RAR message is sent by a BB to its peer with information related to the QoS request and the other parameters of the service. A RAA (Resource Allocation Answer) message containing the answer to the RAR is sent back by the other BB peer. If the required resources are available, the request is propagated recursively through the inter-domain path to the last BB. This last BB returns a RAA message to its immediate downstream BB and the process is continued until it reaches the original BB. This is concluded with an admission of the QoS request. However, we must highlight the fact that SIBBS may accept data flow even if this flow is later rejected in another domain, as can be seen in Fig. 2.

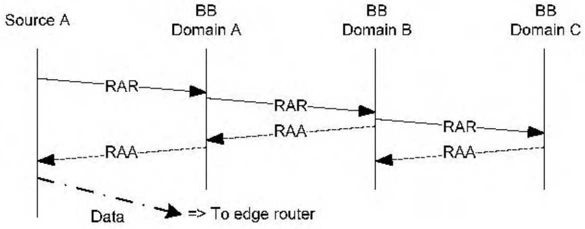

Figure 2. SIBBS signaling between three domains

The other suitable inter-domain communication protocol is COPS-SLS, which is an extension of the COPS for SLS management in a multi-domain environment. It has the same behaviour as SIBBS since a request is propagated from one BB to the other in each domain of the data path. However, it is obviously designed for policy control rather than negotiation and it works in a client-server environment where each BB has the double role of PDP for the upstream domain when the BB sends the request and PEP for the next BB domain.

Given that COPS-SLS only adds a few features compared to SIBBS, such as the renegotiation of the classes of service in the event of admission control failure, and that we want to keep this first implementation of an IPv6 inter-domain signaling as simple as possible, we will use SIBBS. Moreover in [27] SIBBS is proposed for inter-domain signaling when scalability is not a factor.

Therefore, according to the aforementioned work of the DiffServ, Policy and RAP WGs and the architecture proposed by the I2-QBBAC, our implementation of a BB for the centralized management of DiffServ domains will use COPS-PR and SIBBS. Two kinds of PEPs will be managed: the domain access routers (edge routers) and the core routers (Fig. 3). The SLA and SLS may be configured statically at the PDP and the service is then realized by the BB receiving a resource allocation request and configuring the routers at the edges of its domain with the set of parameters for the PHBs and the Traffic Conditioning mechanisms.

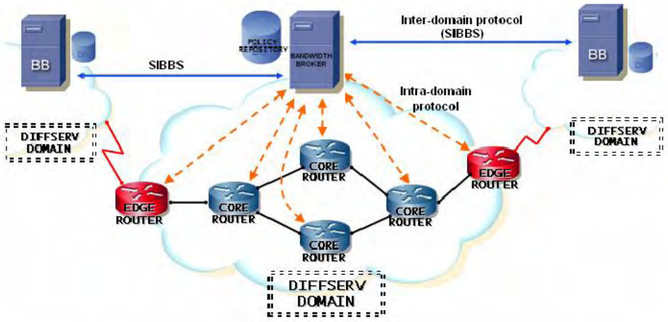

Figure 3. Network architecture defined by the I2-QBBAC

III. Description of the BBv6

We have used the Open Source code of the UNSW [28], in their final 2003 version, as a base of the implementation of our BB (BBv6) [6]. The original BB, developed in JAVA language, supports COPS, COPS-PR and SIBBS and it is based on the definitions of the Policy and RAP WGs and the I2-QBBAC. The BBv6 improves the code by implementing some of the deficits and adding IPv6 support in all the modules. Therefore, our implementation has transformed the original BB into a dual stack device which implements all the requirements to support IPv6, as can be seen in Fig. 4. The COPS-PR and SIBBS implementation, the database and the management console have been reviewed and modified for the addition of the IPv6 support. Furthermore we have modified the SIBBS protocol which we refer to as SIBBSv6.

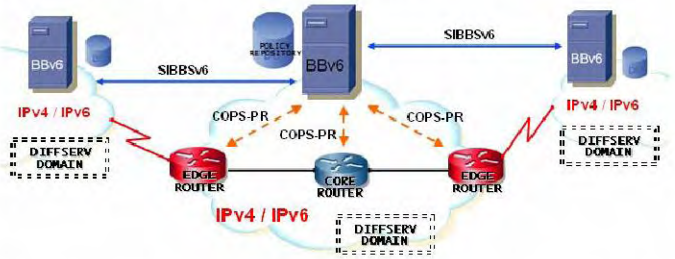

Figure 4. BBv6 network architecture for inter/intra-domain comunication over dual IPv4/IPv6 Domains

A. *Management console and Database modules*

The management console operates with the SLA/SLSs concept [8][10]. Every BB may have many SLAs from different clients and each one may have many SLSs for every client's specific needs which rule the technical parameters (reserved BW, type of service requested, etc.) of the resource reservation at the domain entrance. The new interface allows the management of both of them and the implementation of the edge router support allows the marking of the packets which satisfy the SLS requirements on entrance to the domain.

The database stores the domain policies such as SLA and the network topology and the addresses of the BBs of adjacent domains. PostgreSQL v.7.4.6. is used because its JAVA connector supports IPv6 connections. The SLS are transferred to the PIB for their distribution through COPS-PR to the PEPs, as well core as edge routers, marked by the network topology. Therefore, the relation of routers of the domain allows the installation of the network policies to the routers.

B. *PEP module*

The PEP module installed in every IPv6 DiffServ node is responsible for the configuration of the policies in the PEP's PIB in a computer running GNU/Linux as a router.

In the IPv6 DiffServ networks of the implemented testbed the edge routers are able to support all the Traffic Conditioning mechanisms described in the DiffServ architecture: classification, metering, shaping, policing (dropping) and marking of packets, in addition to queuing. The core interfaces are able to manage marked traffic on the basis of the Behavior

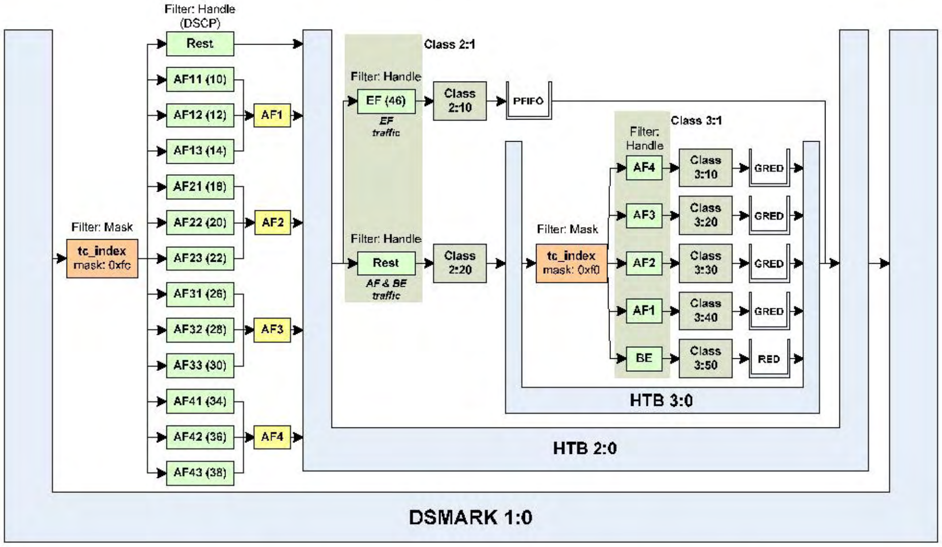

Figure 5. Queuing system of the core routers with GNU/Linux OS

Aggregate classifier. The queuing systems used in these GNU/Linux nodes are mainly the DSMARK and Hierarchical Token Bucket (HTB) queuing disciplines as can be seen in Fig.5, even though PFIFO and virtual queues (for the AFxx sub-classification inside the GRED system) have been also used. HTB permits the definition of relative bandwidths for each class of traffic and accepted bandwidth to borrow from other classes.

The GNU/Linux kernel has been configured for IPv6 support with all the QoS functionalities available to be used in IPv4 as well as IPv6.

Hence, the new PEP module is ready to manage all the PHB traffic types: Expedited Forwarding (EF) [29] for delay sensitive premium traffic, Assured Forwarding (AF) [30] for non-critical priority traffic and BE for the rest of the traffic.

C. *Intra-domain module*

The protocol used for the intra-domain communication is COPS-PR. Besides the support of IPv6 protocol, we implement the whole protocol, including the keep-alive function, the synchronization function for the PDP-PEP disconnection case and the PEP-redirect function in fail case. This latter function also supporting PDP redirection with IPv6 addresses. Support for the PIB defined by the DiffServ WG [31] has also been included in addition to the previously determined RAP WG PIB definition [32], which always supports IPv6. Our PIB completely fulfils them.

D. *Implementation of the inter-domain module*

The inter-domain communication between the BBv6 peers is done on a point-to-point basis. This way a BBv6 is only capable of interacting with its neighbors. To reach further domains the BBv6 will use a predefined BBv6 as a default gateway.

In the SIBBSv6 protocol the resource reservation between domains is now carried out through SLAs and SLSs. Each BBv6 has a unique SLA for each BBv6 peer in its DB. All the SLSs in this SLA will determine the policies at the entrance of the domain for the traffic incoming from the peer domain. Therefore the BBv6 now becomes the “client” and the SLSs between domains will be associated to the same SLA, even if they are from different external clients. This way we prevent confidential client data from being sent to the Service Provider of the peer domain.

We have also modified the signaling process. Now the source client cannot send the data until all BBv6 along the domain path has accepted it. If a domain can not accept the SLS, it sends a Notification Error/Fail message and the process is stopped (Fig. 6). This consequently implies a greater delay but avoids sending data which could be not accepted by a remote BBv6.

Furthermore, the SIBBSv6 protocol has been adapted to support IPv6.

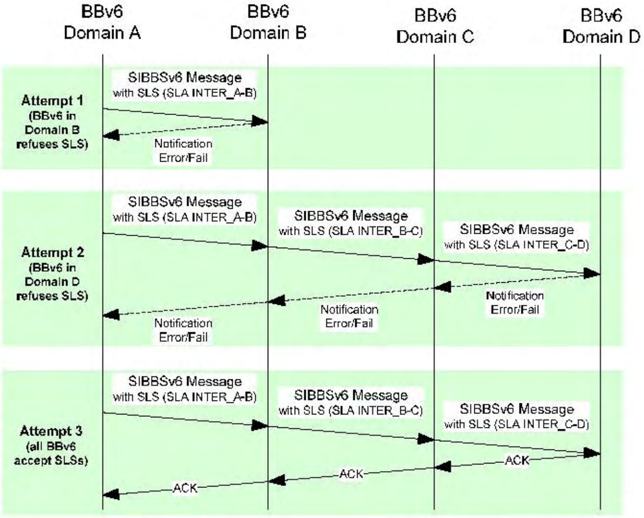

Figure 6. Inter-domain signaling sequence between four domains

IV. Evaluation

The performance of the policy-based architecture has been evaluated through the implementation of a testbed with three IPv6 DiffServ domains (Fig.7). All the computers in the testbed are Intel Pentium IV, 2.8 GHz running Debian GNU/Linux 3.1 with rebuilt Kernel 2.6.9 with all the QoS options. The PEP modules have been installed in the two edge routers and in the core router in each of the three domains.

Therefore, the goals of this inter/intra-domain scenario are to evaluate the correct performance of the BBv6, the IPv6 DiffServ nodes, the COPS-PR protocol implementation and the SIBBSv6 protocol over the IPv6 DiffServ domains.

A. *Testing policies*

As can be seen in Fig. 7, there are three concatenated domains. Therefore according to the SIBBSv6 protocol there will be four inter-domain SLAs.

- The BBv6 in Domain A has an inter-domain SLA named INTER_B-A with the SLSs which manage the policies for the incoming traffic from Domain B at the edge router PEP3A (the source of the traffic could also be in Domain C).
- The BBv6 in Domain B has two inter-domain SLAs named INTER_A-B and INTER_C-B with the SLSs which manage the policies for the incoming traffic from Domain A and C respectively at the appropriate edge routers (PEP1B and PEP3B).
- The BBv6 in Domain C has an inter-domain SLA named INTER_B-C with the SLSs which manage the policies for the incoming traffic from Domain B at the edge router PEP1C (the source of the traffic could also be in Domain A).

Besides the inter-domain SLAs, each BBv6 will have their own client SLAs. In the case of the Fig.7 scenario, these will be:

- Client-X has an SLA in the BBv6 in Domain A with two SLSs (1 and 3 in Fig. 8) for traffic with destination in Domain B (sink-1). Traffic sources src::A, src::B and src::C are from this client.
- Client-Y has an SLA in the BBv6 in Domain A with two SLSs (2 and 4 in Fig. 8) for traffic with destination in Domain C (sink-2). Traffic sources src::D, src::E and src::F are from this client.
- Sink-1 has an SLA in the BBv6 in Domain B.
- Sink-2 has an SLA in the BBv6 in Domain C.

All these policies will be delivered to edge routers by COPS-PR and therefore will have to be mapped in its PIB. To give an idea of how the PIBs are configured we provide the PIB policies to be used in PEP1A (Fig. 8).

In order to implement the traffic conditioner required by DiffServ to support the aforementioned SLSs, the PIB in Fig. 8 will be installed in the edge router PEP1A. For the sake of simplicity we have omitted the PRID fields as well as those which are not applicable and therefore have a zeroDotZero value.

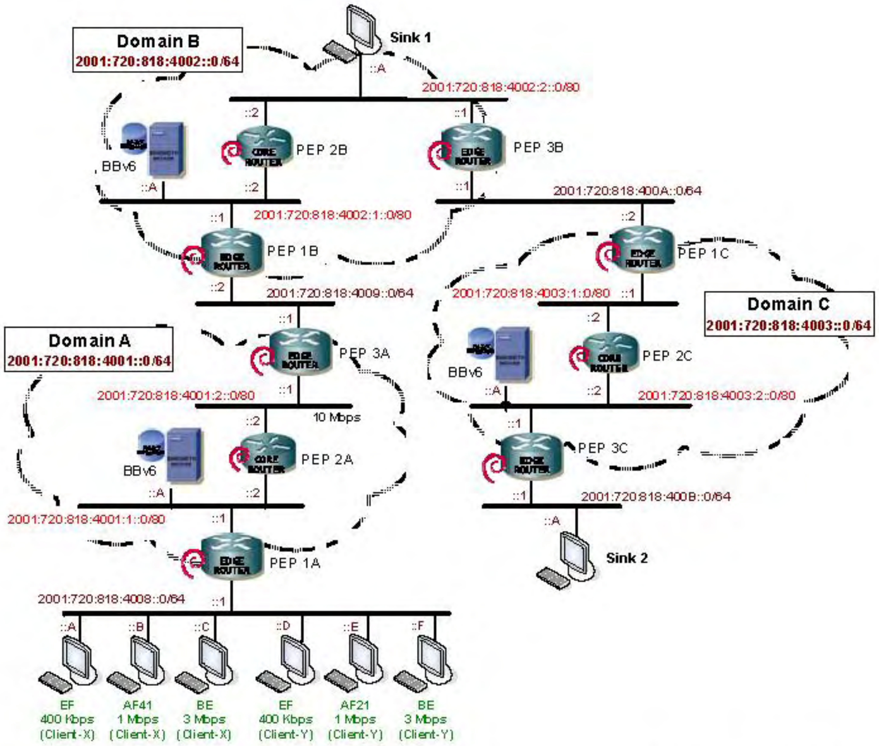

Figure 7. Scenario implemented to evaluate the performance of the inter-domain communication

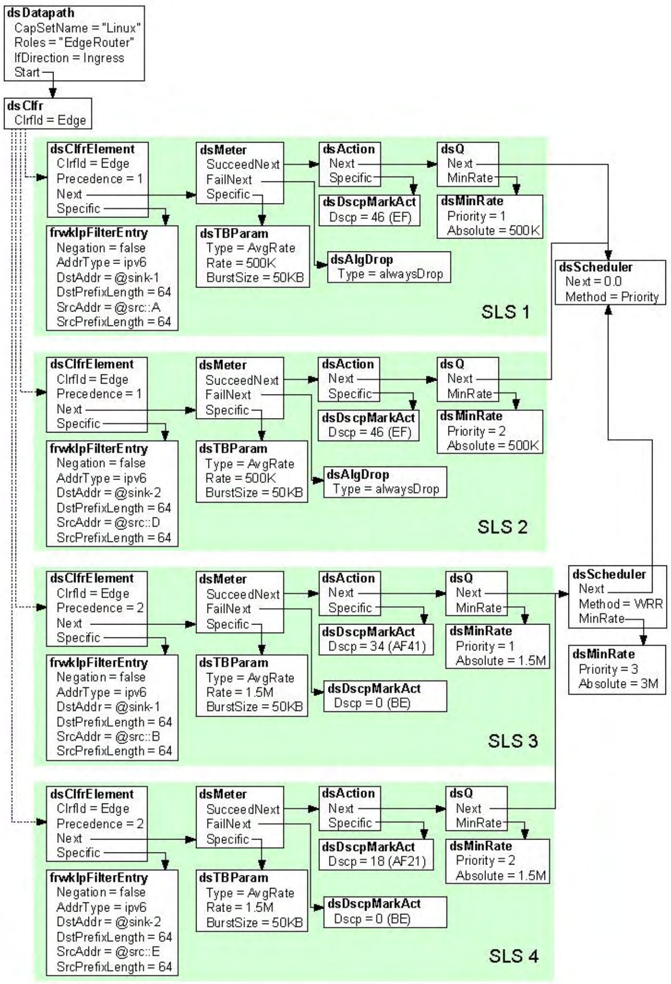

Figure 8. PIB policies of the edge router PEP1A

There are four classifier elements in the PIB, one for each installed SLS with its corresponding IPFilter, which maps the data by identifying the conformance traffic. All four SLSs have destinations belonging to other domains.

- SLS1: EF Class, DSCP=46, src=@src::A, dst=@sink-1, rate= 500 Kbit/s with bursts of 50 KB, out-of-profile action = DROP.
- SLS2: EF Class, DSCP=46, src=@src::D, dst=@sink-2, rate= 500 Kbit/s with bursts of 50 KB, out-of-profile action = DROP.
- SLS3: AF41 Class, DSCP=34, src=@src::B, dst=@sink-1, rate= 1500 Kbit/s with bursts of 50 KB, out-of-profile action = REMARK as BE.
- SLS4: AF21 Class, DSCP=18, src=@src::E, dst=@sink-2, rate= 1500 Kbit/s with bursts of 50 KB, out-of-profile action = REMARK as BE.

On the other hand, the PDP sends other PIB policies to the core routers (Fig. 9). We have used the same policies in the core routers of the three domains. Once they are installed the routers are able to classify packets according to EF and AF PHBs.

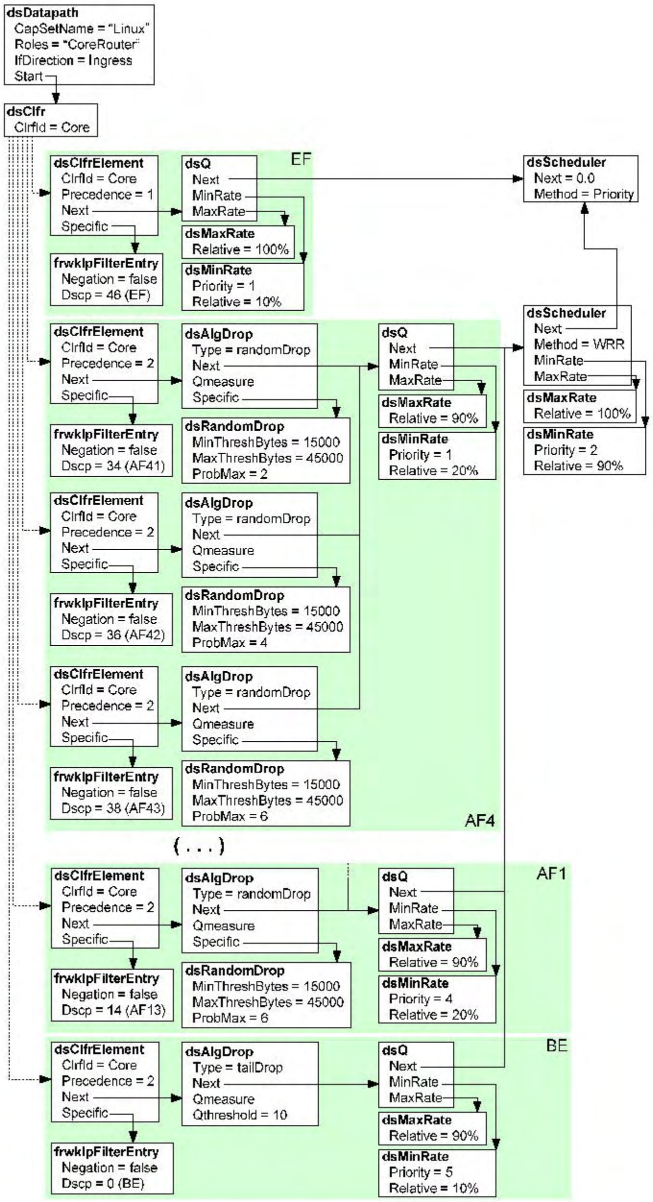

Figure 9. PIB policies of the core routers

These policies state that the EF class has top priority in the packet queuing system. The AF classes append the possibility of using GRED to provide discarding probabilities which will depend on the subclass used. The PIB uses the Random Dropper to define GRED.

- EF class: DSCP 46, 10% BW allocated, 100% allowed if no congestion.
- For each AFx class: 20% BW allocated, 90% allowed to borrow from other classes if no congestion. The GRED discarding probability parameters are:
  - AFx1 drop probability: 0.02
  - AFx2 drop probability: 0.04
  - AFx3 drop probability: 0.06

- BE class: DSCP 0, 10% BW allocated, 90% allowed to borrow from other classes if no congestion. TailDrop dropping algorithm

The mapping of the Hierarchical Token Bucket (HTB) queuing discipline inside the PIB is created through the concatenation of PRC dsSchedulers.

B. *Results*

Six different UDP traffic sources have been configured to carry out the tests: Two EF traffic sources which generate 400 Kbit/s and will be marked at the edge router PEP1A, two AF traffic sources which generate 1000 Kbit/s each and will be marked as AF41 and AF21 at the edge router PEP1A and finally two BE traffic sources which generate 3000 Kbit/s. The traffic has been generated with Iperf 2.0.2. We should point out that with Iperf the datagram size needs to be reduced when using IPv6 addressing 1450 bytes or less to avoid fragmentation.

The link to be congested in this scenario has been implemented with a 10 Mbit/s Ethernet hub in the network 2001:720:818:4001:2::0/80 in DomainA.

Firstly, the three BBv6s initialize all the nodes in their domain. Once the policies for the core routers are set in the BBv6s of the domains, the BBv6s install the policies of the core routers using its corresponding PIB. Then when the administrator configures the new four SLSs in DomainA the BBv6 of this domain checks if it has enough resources available to serve each SLS locally. In case of error or in case of denegation of resources, the SLS is placed in Standby mode and a report is sent to the administrator.

The destinations of the four SLSs are localized in another Domain so BBv6/DomainA has to send SIBBSv6 requests to BBv6/DomainB. The latter BBv6 checks if it can serve the SLSs with the assigned resources to SLA INTER_A-B and with the remaining resources in DomainB. In case of error in any of the SLSs, it will deny the request of the concrete SLS by sending an error/fail notification to BBv6/DomainA which will place the SLS in Standby mode and the administrator will be informed.

The destinations of two of the SLSs are localized in Domain C so BBv6/DomainB sends SIBBSv6 requests to BBv6/DomainC. Then, this BBv6 checks if it can serve the SLSs with the assigned resources to SLA INTER_B-C and with the remaining resources in DomainC. In case of error in any of the SLSs, it will deny the request of the specified SLS by sending an error/fail notification back to BBv6/DomainB which will send an error/fail notification to BBv6/DomainA. BBv6/DomainB will erase the local request but BBv6/ DomainA will place the SLS in Standby mode and the administrator will be informed.

When all the BBv6 along the path have accepted the SLSs, BBv6/DomainC reserves resources in the corresponding SLSs in the SLA INTER_B-C and sends an acknowledgement to BBv6/DomainB. This latter does the same with the SLA INTER_A-B and, finally BBv6/DomainA, on receiving the acknowledgement from BBv6/DomainB, places them in Active mode.

Once the policies have been accepted in the BBv6s of the different domains, the BBv6s will apply the configured SLSs to the edge routers. As stated before, the source client cannot send the data until all BBv6 along the domain path have accepted the related SLS and the policies have been updated at the edge routers.

Figs. 10 and 11 show the initial behavior of the network before the policies have been configured at the edge routers and when these policies are applied in the three BBv6 and distributed to the edge routers for the incoming packet treatment after 30 seconds. The figures show the accumulated dropped packets for each type of traffic in the two sinks belonging to different domains.

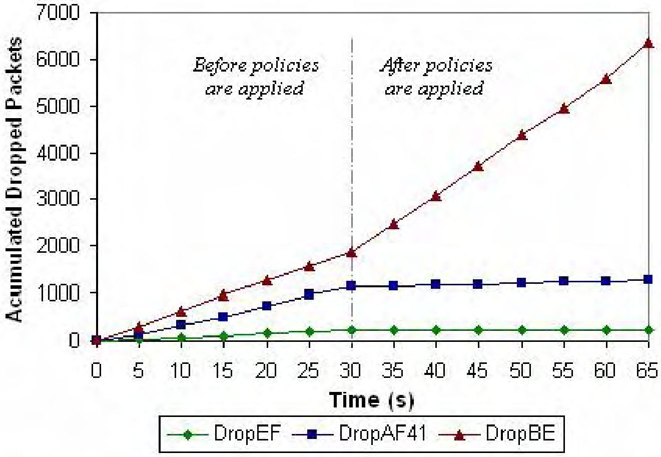

Figure 10. Packets dropped destined to sink-1 before and after policies are applied by the BBv6s

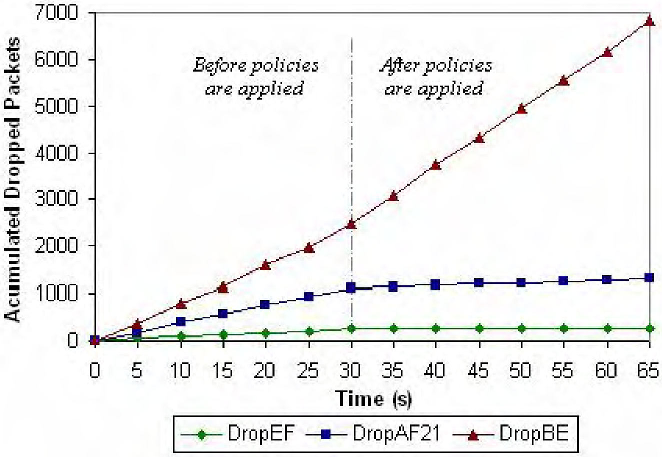

Figure 11. Packets dropped destined to sink-2 before and after policies are applied by the BBv6s

As seen in Table I, once the policies are applied the system stops dropping EF packets, reduces the AF41 packets drop to 4.79 % and the AF21 packets drop to 6.03 % and increases to 33.68 % (average of both sinks) the BE packets drop. These results are coherent to the policy applied since the EF queue has absolute priority and AF41 has a lower discarding probability than AF21.

TABLE I. PACKETS DROPPED IN PERCENTAGE (INTER-DOMAIN)

	Sink 1		Sink 2	
	Before policies are applied	***After policies are applied***	***Before policies are applied***	***After policies are applied***
EF	14.97 %	0 %	16.46 %	0 %
AF41	22.88 %	4.79 %	-	-
AF21	-	-	22.43 %	6.03 %
BE	13.67 %	33.86%	17.5 %	33.50%

V. CONCLUSIONS AND FUTURE WORK

In this paper an enhancement of the BBv6, our bandwidth broker architecture, has been presented, described and evaluated. We have modified the code to incorporate an inter-domain communication protocol. This protocol is a modified version of the SIBBS protocol where the resource reservation is now carried out through SLAs/SLSs and the completeness of the acceptance notification of all the remote BBv6 along the path is now a requirement before sending the data in the signaling process. This new SIBBS version is named SIBBSv6.

Therefore, this system now provides the Operators with a Bandwidth Broker to centrally manage native IPv6 DiffServ domains which permits the delivery of network policies through the COPS-PR intra-domain communication protocol to IPv4 and/or IPv6 domain nodes and it permits inter-domain negotiation through the SIBBSv6.

The BBv6 code has been programmed with JAVA for multiplatform support, it is Open Source and freely available.

A testbed with three IPv6 DiffServ domains has been designed and implemented to evaluate the operation and performance of the architecture and it has been successfully executed. The results show how the BBv6 has successfully distributed the DiffServ policies and they prioritize all the premium traffic and most of the AF traffic although they do not guarantee the latter.

We are currently working on the BBv6 for RSVP and plan to evolve the BB to a PCE towards the management of MPLS TE and MPLS DS-TE networks in the near future.

ACKNOWLEDGMENT

We would like to thank the "Ministerio de Educación y Ciencia" for its support (financial backing) under Grant number TEC2005-08068-C04-04/TCM. We must also thank "Enginyeria i Arquitectura La Salle" (Ramon Llull University) for their encouragement and assistance, especially C. Duz and the NewTech group at the Networking Laboratory for their support throughout the development of this paper and L. Kinnear for the linguistic reviews of the paper.

REFERENCES

[1] University of Tokyo, *USAGI (UniverSAl playGround for Ipv6) Project*, December 2006, [Online]. Available: http://www.linux-ipv6.org

[2] J. Ruiz, A. Vallejo, J. Abella, "IPv6 Conformance and Interoperability Testing", IEEE Symposium on Computers and Communications (ISCC'05), 10th ISCC, pp. 83-88, June 2005.

[3] A. Vallejo, J. Ruiz, A. Zaballos, J. Abella, J.M. Selga, "State of the art of IPv6 conformance and interoperability testing", IEEE Communications Magazine, 2007. In press.

[4] K. Nichols, V. Jacobson, L. Zhang, "A Two-Bit Differentiated Services Architecture for the Internet", IETF RFC 2638, July 1999.

[5] IETF Path Computation Element (PCE) Working Group Charter, December 2006, [Online]. Available: www.ietf.org/html.charters/pce-charter.html

[6] A. Vallejo, A. Zaballos, J. Abella, J.M. Selga, C. Duz, "Performance of a policy-based management system in IPv6 networks using COPS-PR", ICN 2007, April 2007. In press.

[7] IETF Differentiated Services (DiffServ) Working Group Charter, December 2006, [Online]. Available: www.ietf.org/html.charters/OLD/diffserv-charter.html

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

[9] X. Xiao and L. Ni, "Internet QoS: A Big Picture", IEEE Network, pp. 8-18, March/April 1999.

[10] D. Grossman, "New Terminology and Clarifications for DiffServ", IETF RFC 3260, April 2002.

[11] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", IETF RFC 2474, December 1998.

[12] IETF Policy Framework (Policy) Working Group Charter, December 2006, [Online]. Available: www.ietf.org/html.charters/OLD/policy-charter.html

[13] IETF Resource Allocation Protocol (RAP) Working Group Charter, December 2006, [Online]. Available: www.ietf.org/html.charters/OLD/rap-charter.html

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

[15] K. Chan, et al., "COPS Usage for Policy Provisioning (COPS-PR)", IETF RFC 3084, March 2001.

[16] K. McCloghrie, et al., "Structure of Policy Provisioning Information (SPPI)", IETF RFC 3159, August 2001.

[17] IETF Configuration Management with SNMP (snmpconf) Working Group Charter, December 2006, [Online]. Available: www.ietf.org/html.charters/OLD/snmpconf-charter.html

[18] S. Waldbusser, J. Saperia, T. Hongal, "Policy Based Management MIB", IETF RFC 4011, March 2005.

[19] R. Neilson, J. Wheeler, F. Reichmeyer, S. Hares, "A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment", v.0.7, Internet2 Qbone BB Advisory Council, August 1999.

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

[21] P. Pan, E. Hahne, and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for interdomain Signaling", Journal of Communications and Networks, Vol. 2, No. 2, June 2000.

[22] S.I. Maniatis, E.G. Nikolouzou, I.S. Venieris, "QoS Issues in the converged 3G Wireless and Wired Networks", IEEE Communications Magazine, Vol. 40, No. 8, pp. 44-53, August 2002.

[23] J. Chen, A. McAuley, V. Sarangan, S. Baba, and Y. Ohba, "Dynamic Service Negotiation Protocol (DSNP) and wireless DiffServ", IEEE International Conference on Communications (IEE ICC 2002), Vol. 2, April 2002.

[24] R. Bless, "Dynamic Aggregation of Reservations for Internet Services", Telecommunication Systems, Vol. 6, No. 1, pp. 33-52, 2004.

[25] T. M. T. Nguyen, N. Boukhatem, Y.G. Doudane, G. Pujolle, "COPS-SLS: A Service Level Negotiation Protocol for the Internet," IEEE Commun. Mag., vol. 40, no. 5, pp. 158-165, May 2002.

[26] S Salsano, E Sangregorio, M Listanti, "COPS DRA: a protocol for dynamic Diffserv Resource Allocation.", Joint Planet-IP NEBULA workshop, 2002.

[27] P Nanda P, AJ Simmonds, K Rajput: "*Policy Based Network Architecture in Support for Guaranteed QoS*", ITPC 2003 International Conference on Information Technology: Prospects and Challenges 2003, Kathmandu, Nepal, May 2003

[28] S. Jha, M. Hassan, "Implementing bandwidth broker using COPS-PR in Java", Local Computer Networks (LCN 2001), 26th Annual IEEE Conference on, pp: 178-179, November 2001.

[29] B. Davie, et.al., "An Expedited Forwarding PHB (Per-Hop Behavior)", IETF RFC 3246, March 2002.

[30] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", IETF RFC 2597, June 1999.

[31] K. Chan, R. Sahita, S. Hahn, K. McCloghrie, "Differentiated Services Quality of Service Policy Information Base", IETF RFC 3317, March 2003.

[32] R. Sahita, Ed., S. Hahn, K. Chan, K. McCloghrie, "Framework Policy Information Base", IETF RFC 3318, March 2003.