Network Services Deployment for QoS provisioning in a multi-layer DiffServ Architecture

E. Nikolouzou, S. Maniatis, P. Sampatakos, H. Tsetsekas, I. Venieris
Department of Electrical and Computer Engineering
National Technical University of Athens
9 Iroon Politechniou, 15773 Zografou, Athens
GREECE
{enik, sotos, psampa, htset}@telecom.ntua.gr, ivenieri@cc.ece.ntua.gr

Abstract: - The Internet evolution delineated during the last years has necessitated the need of quality of service differentiation among IP flows that expose different characteristics regarding bandwidth, delay, packet loss, and other QoS-relative parameters. This paper describes in brief the specification and implementation of a layered architecture, which achieves the aforementioned requirements through the definition of specific modules that cater for all the aspects of quality: user-oriented reservations, selection of the appropriate traffic class, admission control, router setup, resource management and measurements. The main innovative feature of the architecture is the definition of the Resource Control Layer that resembles a distributed Bandwidth Broker and offers a feasible way to implement admission control at the edges and manage the resources of the network. The paper focuses on the definition of specific Network Services targeting different kinds of applications. They are implemented within the network through the appropriate Traffic Classes, which properly set various parameters of the network routers.

Key-Words: - Quality of Service, Differentiated Services, Resource Management, Network Services

1 Introduction

The Internet Protocol (IP) is a best effort protocol that lacks any QoS mechanisms. The definition of the Integrated Services (IntServ) [1] architecture was the first step for the introduction of QoS in the Internet. IntServ makes use of the Resource Reservation Protocol (RSVP) [2] in order to reserve resources at every network node on the path from the sender to the receiver. However, the use of RSVP for single-flow reservations as well as the constant exchange of RSVP messages in the network raised concerns on the scalability of the architecture. Differentiated Services (DiffServ) [3] aimed to provide a scalable and efficient architecture by supporting relative prioritization of IP traffic. Suitable buffers are created in every network node in a DiffServ domain, with each one configured to serve the QoS requirements (in terms of delay, jitter etc.) of a specific Traffic Class. The introduction of the Bandwidth Broker [4, 5] enhanced DiffServ with an entity that centrally manages the resources and controls the nodes of a domain.

However, there is still a gap between applications requiring QoS and the DiffServ network. A means is needed for the network to form Network Services out of Per-Hop Behaviors (PHBs) [3] or Per-Domain Behaviors (PDBs) and for users or applications to request QoS for a specific IP flow. Moreover, there is a need for an entity that will control an administrative domain and provide dynamic QoS in a scalable and efficient way.

This paper briefly presents a layered architecture for QoS provision in heterogeneous networks. A new layer, the Resource Control Layer (RCL) is introduced on top of a DiffServ network. The RCL can be viewed as a distributed Bandwidth Broker: it manages an administrative domain by controlling its resources, configuring its network nodes and performing admission control for new QoS traffic. However, the RCL extends the Bandwidth Broker concept by providing dynamic end-to-end QoS as well as a concrete reservation interface to users and applications.

This paper focuses then on the Network Services (NSs) that our architecture supports. NSs are defined as the overall treatment of a customer's traffic across a particular domain or end-to-end. In order to implement the NSs, some network's mechanisms are defined, which are the Traffic Classes (TCLs).

In the rest of the paper we first briefly present the proposed architecture and the entities that compose the RCL. In section 3, we focus on the concept of Network Services and Traffic Classes. In Section 4, a simulation scenario and results are presented, proving the correctness of the PCBR network service for the provision of end-to-end QoS to a voice flow. Finally, section 5 discusses the conclusions and the plans for future work.

2 Network Architecture

The proposed architecture consists of two functional areas: the data plane that is responsible for transmitting IP packets and an overlay control plane, namely the Resource Control Layer (RCL) that is based on the Bandwidth Broker concept.

However, the classical bandwidth broker (BB) architecture proposes a concentrated approach where one bandwidth broker is responsible for an administrative domain. In order to overcome the scalability problems arising from the centralized nature of the classical BB, the RCL is designed as a *distributed* Bandwidth Broker.

The RCL is responsible -among others- for the management of the available resources, the per flow policy-based admission control, the configuration of the edge devices (EDs), the monitoring of the network and the interaction with the host computers.

As depicted in Fig. Fehler! **Unbekanntes Schalterargument.**, the three key components of the RCL are: the Resource Control Agent (RCA), the Admission Control Agent (ACA) and the End-User Application Toolkit (EAT). These modules are briefly explained in the following subsections.

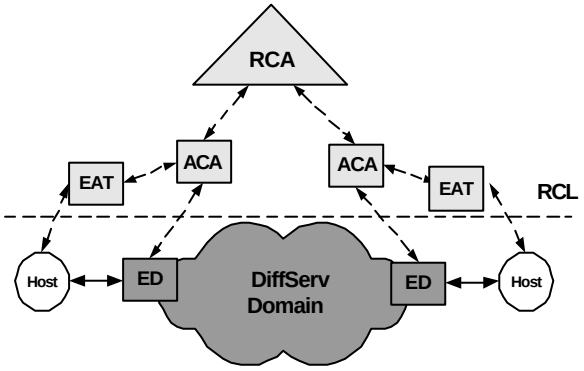

Fig. Fehler! Unbekanntes Schalterargument.**. RCL structure and main interactions**

2.1 The Resource Control Agent

The Resource Control Agent represents the ultimate principle of the domain concerning the management of resources.

It has been examined that the number of user requests is dramatically increasing and a standalone management entity could not perform well under these conditions. Therefore, in order to simplify the task of the RCA to handle the network resources efficiently, the network is divided into sub-areas that form a tree structure, where each sub-area is assigned its own resources. The network administrator estimates these resources according to traffic load forecasts and/or results retrieved by a measurement-based platform.

Therefore, the RCA is divided in logical entities (Resource Pools, RP) and each one of them is assigned the task of managing the resources of a sub-area. The RCA is based on the hierarchical structure depicted on Fig. Fehler! **Unbekanntes Schalterargument.**.

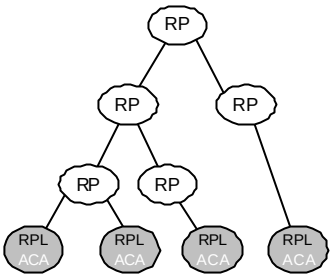

Fig. Fehler! Unbekanntes Schalterargument.**. Hierarchical Structure: Resource Pool Concept**

Every node of the tree has none or many children and exactly one father, except from the root node that has no father. In addition, each leaf of the tree structure (Resource Pool Leaf, RPL) is associated to one Access Control Agent (ACA). During the start-up configuration procedure, the RPs/RPLs are assigned their initial resources, which are provided by a database handled by the network administrator. Since the initial resources may not reflect the actual traffic load of each sub-area, the RPs/RPLs should be able to adjust resource assignments to real traffic conditions [6], which are difficult to be forecasted and may change during time.

2.2 The Admission Control Agent

An Admission Control Agent (ACA) administrates each edge device of the network. The ACA mainly performs three major tasks, including user authentication and authorization, reservation handling, and admission control

Firstly, in order to contact the ACA, the user must be authenticated. Then, a reservation request, which specifies the traffic specifications of the new flow, can be made. Secondly, the user authorization for that type of request is checked and a reservation state is created. As aforementioned, the policies and the admission control decisions are made only at the edges of the network, therefore the corresponding ingress and egress points (ingress-egress ACAs) of the flow are identified and the local resources are checked to ensure that the new flow can be accommodated. If the resources are insufficient, more resources can be requested from the RCA.

2.3 The End-User Application Toolkit

An essential part of any QoS network architecture is the ability of the users to communicate their requirements regarding quality towards the network. The End-User Application Toolkit (EAT) mediates between end-users or applications and the network. The major role of the EAT is to pass user reservations to the ACA. The reservations concern individual IP flows and specify the selected network service and the corresponding traffic class profile. Moreover, they may specify the desired time and date that the reservation will take place. End-user reservations can be placed through a Web-based graphical user interface (the QoS Console), while application reservations can be requested through an Application Programming Interface (API). The latter is a simple API that is not intended to replace well-established protocols, but to offer the most flexible way to take advantage of the new QoS architecture. It is essential that the EAT will be able to support standardized APIs, like RSVP, or the emerging SIP-based QoS extension API [7].

3 Network Services & Traffic Classes

Our architecture provides quality of service guarantees to the users by offering a number of transport options for user IP traffic. These are called Network Services, which are defined as the overall treatment of a customer's traffic across a particular domain, across a set of interconnected DS domains, or end-to-end [8, 9]. Services are constructed by applying traffic conditioning [10] to create aggregates that experience a known Per-Hop-Behavior (PHB) at each node within the DS domain. Services can be clearly categorized as qualitative or quantitative depending on the type of performance parameters offered.

In the proposed architecture five Network Services (NS) have been defined in order to provide service guarantees to different applications: Premium Constant Bit Rate (PCBR), Premium Variable Bit Rate (PVBR), Premium MultiMedia (PMM), Premium Mission Critical (PMC), Best Effort (BE)

Applications can be grouped into this relatively small number of services, with the applications in each service having similar requirements on the network in order to perform effectively.

The PCBR is for CBR and VBR applications with low bandwidth flows. These applications have low delay and delay variation requirements, although are tolerant to some packet loss. In addition, they should have small packets so as not to provoke long transmission delays. IP telephony is the basic application supported by this NS. For good voice quality the delay should be less than 150msec for the 99% of in-profile packets and the packet loss should be less than 10^{-8}.

The PVBR is appropriate for unresponsive VBR sources with medium to high bandwidth requirements and low delay and packet loss requirements, even though greater than the ones of the PCBR. An end-to-end delay less than 200msec and a packet loss less than 10^{-4} are guaranteed. The intention is to separate these unresponsive flows from responsive flows (PMM) in order to inhibit the unresponsive VBR flows to steal away the entire capacity from the responsive flows. Additionally, it is reasonable to separate possibly high bandwidth VBR flows from the low bandwidth VBR and CBR flows in PCBR, since peak rate allocation is inefficient for high bandwidth VBR flows. An interactive video transmission would be a possible application supported by this network service.

The PMM is expected to carry a mixture of TCP and non-TCP traffic. These flows will require a minimum bandwidth, which must be delivered at a high probability. Independently of the transport protocol, flows are expected to implement some kind of congestion control mechanism and their aggressiveness should be similar to the one of TCP. That means that all flows are assumed to be roughly TCP-friendly [11]. Applications such as video/audio streaming and ftp can be delivered by this NS. The drop probability for in-packets should be very low (less than 10^{-3}) while out-of-profile packets do not experience any QoS guarantees.

The PMC service supports mainly transactions and database queries. Thus, flows of the PMC are non-greedy, have short lifetimes, low bandwidth requirements and roughly homogeneous congestion control (TCP). The low bandwidth property allows overprovisioning of this service in the network, enabling minimal loss probabilities for in-profile packets (less than 10^{-6}) and small queuing delays. Besides, negligible loss probability and low delay are important requirements of this service.

3.1 Traffic Classes

In order to implement the NSs that are offered by the network provider to the customer, some network's mechanisms are defined, which are the Traffic Classes (TCLs). The TCLs are introduced in the network and are known to ACA, which associates the NSs to TCLs. An NS can be composed of more than one TCL. In the proposed

architecture there is a one-to-one correspondence and therefore five TCLs are introduced: TCL1, TCL2, TCL3, TCL4 and TCL STD which correspond to PCBR, PVBR, PMM, PMC and BE.

A TCL is defined as a composition of a set of admission control rules, a set of traffic conditioning rules and a PHB [9]. A PHB is implemented by means of a queue management algorithm (Random Early Detection (RED), Weighted-RED (WRED)) and a scheduling algorithm (First-In-First-Out (FIFO), Priority Queuing, Weighted-Fair Queuing (WFQ), Class-Based Queuing (CBQ)).

For each of the five TCLs a separate queue is maintained at the router output ports scheduled with WFQ [12]. WFQ minimizes the interactions between TCLs, and provides flow isolation desired for providing the different QoS requirements of each TCL. A weight is assigned to each TCL according to the bandwidth dedicated to each traffic class and bandwidth borrowing between traffic classes is supported. The queues created are managed by different queuing strategies.

In the next paragraphs we describe the traffic conditioning and queue management mechanisms used for the implementation of each traffic class.

The use of a token bucket meter and dropper is proposed for the traffic conditioning mechanisms for the TCL1. The token bucket is configured with a token generation rate r and a bucket size b, which are set as:

r = *Peak Rate in bits/sec (PR) of the flow* (1)

$b = x * M_1$, (2)

where x is a fixed value chosen by the network operator in the range of [1,5]; a possible value could be x=1. A larger value of x would allow a small amount of burstiness. M is the Maximum Policed Unit for this traffic class, and a possible value is M_1=256bytes. The token generation rate specifies the maximum rate at which sender can transmit its packets and the bucket size defines the degree of burstiness of the traffic.

Packets that do not find enough tokens in the bucket are dropped. Packets of TCL1 are enqueued in a FIFO drop-tail queue.

The traffic conditioning mechanism appropriate for TCL2 is a dual token bucket as meter and dropper. The token generation rate and the bucket size is set for each token bucket as:

r_1 = *Sustained Rate in bits/sec(SR) of the flow* (3)

b_1 = *Bucket Size for SR in bytes(BSS)* (4)

r_2 = *PR* (5)

$b_2 = x * M_2$, (6)

where x is a fixed value chosen by the network operator in the range of [1,5] and M_2 can be set to 1500bytes. The dual token bucket works in the following way: if there are enough tokens in the first *and* in the second bucket to accommodate a packet, then it is marked as in-profile. Otherwise the packet is dropped. The intention is to limit the sender's traffic in order to be conformant to the profile of the first token bucket (SR, BSS), while an amount of burstiness is allowed by the second token bucket. The second token bucket defines the allowed maximum rate (PR), at which the sender can transmit its packet. Packets of TCL2 are enqueued in a single FIFO drop-tail queue.

The traffic conditioning mechanism for TCL3 is realized with the use of a single token bucket as a meter and marker. The token bucket will police sustained rate, and it will be configured as:

r = *SR of flow* (7)

b = *BSS* (8)

The flows conforming to this profile will be marked as in-profile otherwise they will be marked as out-of-profile. The bucket size should be very high to satisfy the bursty nature of TCP traffic. In this way the TCP traffic can utilize in a great degree the token generation rates [11]. In addition, the bucket size depends on the per-flow bandwidth * RTT product for TCP flows. The M_3 for this traffic class can be set to 1500bytes.

WRED [13, 14] with two sets of min_threshold, max_threshold, max_propability, one for the in-profile and one for out-of-profile packets is employed as queue management algorithm for TCL3. For this TCL two DSCPs are used, one for in- and one for out-of-profile packets.

For TCL4, a dual token bucket for traffic conditioning and WRED with two sets of min_threshold, max_threshold, max_propability at router output ports are proposed in order to discriminate out-of-profile packets against in-profile packets.

The first bucket works as a sustained rate policer while the second bucket works as a peak rate policer. The two token buckets are configured as:

r_1 = *SR of flow* (9)

b_1 = *BSS* (10)

r_2 = *PR* (11)

$b_2 = x * M_4$, (12)

where x is a fixed value chosen by the network operator in the range of [1,5] and M_4 can be set to 1500bytes.

The token bucket is operated as a meter and marker i.e. out-of-profile packets may enter the net. A packet that requires fewer tokens than available in the first bucket *and* in the second bucket is marked as in-profile. Otherwise, the packet is marked out-of-profile and forwarded into the net.

The token rate should be small in order to disable greedy sources to transmit in-packets with a high rate into the net. The bucket size should be large enough to allow several back-to-back packets to enter the net without being marked as out-of-profile. For this TCL two DSCPs are used, one for in- and one for out-of-profile packets.

For TCL STD no admission control and traffic conditioning is required. At router output ports, best effort packets are enqueued in a single FIFO queue that is scheduled by WFQ. The recommended queue management algorithm is RED.

4 Simulation Scenarios

In order to verify the quality of service provided by the proposed architecture a scenario is provided where the PCBR network service is tested. The corresponding mechanisms (CAR, WFQ) are configured in routers as appropriate for this network service and the end-to-end delay and packet loss are measured.

The Opnet 7.0 simulation tool is used for the simulations. The measurements from the proposed architecture are compared with their corresponding ones in a network that uses the FIFO scheduling discipline (the current Internet implementation). The network used is described below.

The core network consists of two routers. There are also four workstations that send voice traffic with a high priority and three workstations that send best effort traffic. The voice traffic is composed of flows, which use the G.711 voice encoder. For the Best Effort traffic, video conferencing traffic with flows of 202kbps is used. The link between the two routers is considered as bottleneck with capacity of 1.554Mbps (DS1), while all the other links are 44.736Mbps (DS3).

Under the proposed architecture, the voice is subject to admission control with the use of CAR, which is configured as: *rate limit* equal to the maximum allowed rate for the PCBR and *normal burst size* equal to the maximum policed unit of the PCBR (M). Packets not conformant to this profile are dropped.

The output port of Router 1 is configured with a WFQ with two queues, one for the best effort traffic and the other for the high priority with weights 70 and 30 accordingly. That means that voice can occupy up to the 30% of the bottleneck (466kbps), while the best effort traffic can occupy maximum up to 70% of the bottleneck (1088kbps). The voice is considered to utilize continuously the 30% while the best effort traffic varies from 606kbps–2424kbps, meaning from 38.5% - 155%.

When the FIFO scheduling algorithm is used, the same variations in traffic are considered. No scenarios where the voice traffic varies are described here, since we are interesting only from the point where congestion starts.

4.1 Simulation Results

A number of simulations have been conducted in order to measure the end-to-end delay and the packet loss for the BE and voice traffic. In all simulations after 100secs of execution the value of delay for voice was stabilized. That value is actually depicted for each simulation in the figures below.

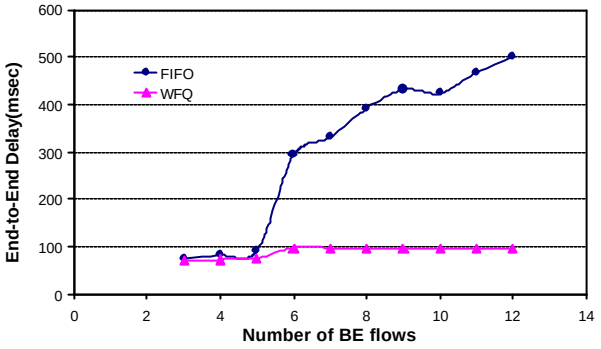

Fig. Fehler! Unbekanntes Schalterargument.**. End-to-End delay (msec)**

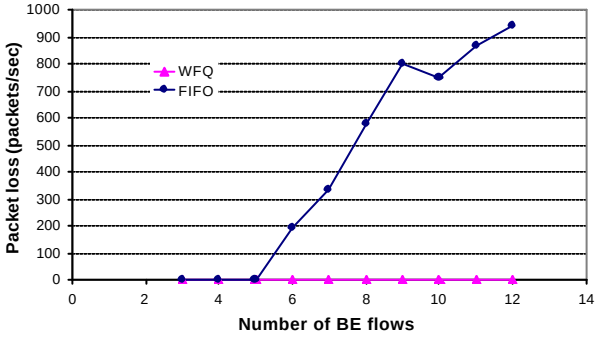

Fig. Fehler! Unbekanntes Schalterargument.**. Packet loss for Voice traffic**

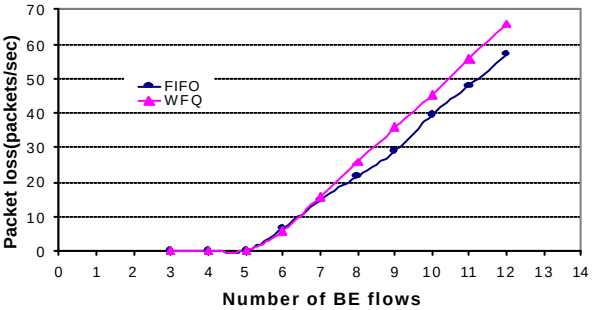

Fig. Fehler! Unbekanntes Schalterargument.**. Packet loss for BE traffic**

From the above figures, it is obvious that better performance is achieved under the proposed architecture. In Fig. Fehler! **Unbekanntes Schalterargument.** the end-to-end delay for voice under the CAR-WFQ implementation is 70.6msec for 3 flows of BE traffic (606kbps) and increases up to the 98msec for 6 flows of BE traffic, which correspond to 1.212kbps, where the congestion starts. It is also observed that the end-to-end delay remains to 98msec independently of the increase of BE traffic from 6 to 12 flows (606kbps – 2.424kbps). That actually verifies the fairness and flow isolation provided by the WFQ. On the contrary, under the FIFO implementation the end-to-end delay increases continually, and from 75msec for 606kbps BE traffic reaches 293msec when the congestion starts (1.212kbps). When the BE traffic is 2.424kbps, the end-to-end delay obtains a really great value (502msec).

In Fig. Fehler! **Unbekanntes Schalterargument.** the packet loss for voice is continually zero under the CAR-FIFO implementation, while it is really high under FIFO. This is justified from the fact that FIFO does not guarantee any bandwidth to neither application, so when the BE traffic increases it occupies a greater portion of the bandwidth in the bottleneck link causing a higher packet loss to the voice traffic. On the other hand, the packet loss for the BE traffic in Fig. Fehler! **Unbekanntes Schalterargument.** is less under FIFO than WFQ. This is explained since under WFQ a specific portion of the bandwidth is guaranteed for each traffic class, while under FIFO the excess BE traffic storms the bottleneck link.

5 Conclusion and Future Plans

In this paper we have presented a distributed architecture for dynamic end-to-end QoS provision in the Internet. The Resource Control Layer is composed of three logical entities that manage a DiffServ domain and reserve resources on request of users and applications. Our approach aims to be more scalable than in the case of a simple Bandwidth Broker, but also to provide users and applications with a means to leverage the services provided by the network. The concepts of the Resource Control Layer are currently tested and verified in laboratory trials and will be tested later in trials involving real users and Internet Service Providers.

At this point our architecture supports only intra-domain reservations. We plan to extend the concept of the Resource Control Layer, so that it covers also reservations that span over multiple administrative domains.

Moreover, the RCL monitoring mechanism will be extended to support error notification and dynamic re-configuration of resource pools, and network devices.

Acknowldgement

This work was performed in the framework of IST Project AQUILA (Adaptive Resource Control of QoS Using an IP-based Layered Architecture - IST-1999-10077) funded in part by the EU. The authors wish to express their gratitude to the other members of the AQUILA Consortium for valuable discussions.

References:

[1] R. Braden et al., Integrated Services in the Internet Architecture: an Overview, *RFC 1633*, June 1994

[2] J. Wroclawski, The Use of RSVP with IETF Integrated Services, *RFC 2210*, September 1997

[3] S. Blake et al., An Architecture for Differentiated Services, *RFC 2475*, Dec. 1998

[4] K. Nichols et al., A Two-bit Differentiated Services Architecture for the Internet, *RFC 2638*, July 1999

[5] R. Neilson et al., A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment, *Internet2 Qbone BB advisory Council*, Aug 1999.

[6] E. Nikolouzou, G. Politis, P. Sampatakos, I.S. Venieris, An Adaptive Algorithm for Resource Management in a Differentiated Services Network, *ICC2001*, June 2001

[7] W. Marshall et al., SIP Extensions for Resource Management, *draft-ietf-sip-manyfolks-resource-00*, November 2000.

[8] Y.Bernet, et al., A Framework for Diff Services, *draft-ietf-diffserv-framework-02*, Feb. 1999

[9] K. Nichols, et al., Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *RFC 2474*, Dec 1998

[10] Y. Bernet et al., Requirements of DiffServ Boundary Routers, *draft-bernet-diffedge-01.txt*, Nov 1998

[11] F.Azeem et al., TCP-Friendly Traffic Conditioners for Differentiated Services, *draft-azeem-tcpfriendly-diffserv-00*, March 1999

[12] J. Bennett, H.Zhang, Hierarchical packet fair queueing algorithms, *SIGCOMM*, August 1996, pg.143—156.

[13] V. Firoiu, M. Borden, A Study of active Queue Management Congestion Control, *Infocom 2000*, March 2000

[14] S. Floyd et al., Random Early Detection Gateways for Congestion Avoidance, *IEEE/ ACM Transaction on Networking*, Aug 1993