A transsignaling strategy for QoS support in heterogeneous networks

Diogo Gomes, Pedro Gonçalves, Rui L. Aguiar

Universidade de Aveiro/Instituto de Telecomunicações, 3800-193 AVEIRO, Portugal

{dgomes, pasg}@av.it.pt, ruilaa@det.ua.pt

Abstract. The increasing usage of multiple signalling mechanisms, with associated QoS extensions, creates several problems to commercial data networks. New and scalable approaches are required for the network operator to support this diversity. This paper discusses a highly flexible, scalable architecture for processing QoS Admission Control in public networks. The architecture relies on the cooperation of two different entities, an agent and a manager, with fully distributed implementation, and able to perform the required signalling, authorization, and admission control decisions. If required, the agent is capable of interfacing with different signalling mechanisms. Early implementation conclusions are also presented. This architecture is capable of operating with multiple QoS frameworks, with minimal added overhead.

Introduction

Internet traffic is increasing at an unprecedented rate as Internet-driven service demand grows. New applications require higher bandwidths and are often quality sensitive. Differentiated traffic treatment is expected for better management of available resources. For this, the Internet Engineering Task Force (IETF) proposed models to support QoS requirements, such as the Differentiated Services (DiffServ) and the Integrated Services (IntServ)/RSVP frameworks. Furthermore, other QoS-related signaling proposals appeared, associated with protocols oriented towards multimedia communications [1].

The IntServ architecture [2] was proposed in order to give QoS guarantees to a specific flow between a source and a destiny. Unfortunately, it presents a severe scalability problem in large networks. On the other hand, the DiffServ [3] framework solves the scaling problem aggregating the traffic flows with similar QoS requirement in Classes of Service (CoS) but does not provide, by itself, end-to-end QoS guarantees, and just provides different treatments to this CoS-marked traffic. In DiffServ scenarios, end-to-end QoS guarantees can be achieved by more complex control strategies: adding a Bandwidth Broker [4] to DiffServ networks; or using hybrid networks, with an access network supporting IntServ, and a transport network DiffServ-aware. Implementations [5][6][7] already exist for these scenarios but they suffer from various problems: lack of flexibility (not able to adapt to new signalling protocols); scalability problems (centralizing the QoSBroker functions in one machine); or extreme inefficiency (carrying IntServ information over the DiffServ network, without any impact on network control).

For an ISP, this QoS multiplicity poses severe difficulties, which are compounded by the extensive requirements of the “new multimedia services”. The operator network will have to adapt to the fact that some applications do not explicitly signal QoS requirements, although they require QoS assurances for proper behaviour, while other applications use complex protocols to negotiate their QoS-related needs. Furthermore, an operator usually has to handle customers with quite different relevance, and it would be preferable to provide differentiated service according to the type of user. This is the target environment for future networks [8], where QoS support will be widespread, and provided in a diversity of situations. In this “4G wireless world” context, the classic Internet paradigm (“keeping all intelligence in the edges”) cannot be maintained, as operators will aim to control network usage. Notice also that the recent security concerns already forces operators to intervene and monitor network traffic.

The next section describes a proposed network architecture for supporting QoS, able to handle this multiplicity of requirements, using a transsignalling unit at the edges. Some potential application scenarios of this architecture are presented in “Transisgnalling usage”, and key conclusions, based on current implementation, are presented in the final section.

An advanced QoS Control architecture

For scalability reasons QoS support at the network level will require DiffServ enabled core networks. Our improved QoS architecture (Fig. 1), based on [9], assumes such a network, and defines specific entities for signalling and control. Three entities are defined, an AAAC server responsible for contract level QoS issues, an AQMUA (Advanced QoS Manager of Universidade de Aveiro) which is mainly a QoS Broker [4] with added functionalities, and SPAAQE (Signalling Processing, Access Authorization and QoS Enforcement) units, an advanced entity lying in access routers, that provides advanced signalling and QoS processing.

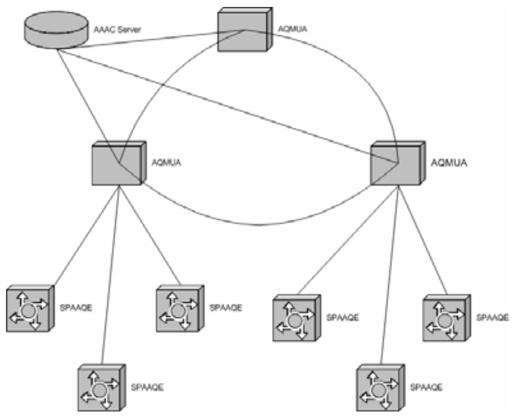

Fig. 1. Proposed QoS-Network Architecture

Each AAAC defines an administrative region for providing a common set of QoS services. Each AQMUA defines an “autonomous” region, independent in its capability of allocating resources, and controlling its associated edge devices (SPAAQE entities). The multiple AQMUAs act as a distributed overlay network for internal signalling, capable of interchanging QoS related information in a fast and simple way.

AQMUA

The Advanced QoS Manager of Universidade de Aveiro is the architecture element responsible for managing end-to-end QoS in its “autonomous” region. The AQMUA is responsible for performing global network management, keeping QoS levels per class; making admission control decisions; keeping information on network topology, as well as information about the neighbor network architectures; exchanging network information with SPAAQE entities; supporting SPAAQE in traffic conditioning decisions, feeding proper queue parameters to that entity and keeping accounting information of user network usage. For doing this, each AQMUA has detailed knowledge of its domain topology and receives reports from its associated SPAAQE entities.

Fig. 2 shows AQMUA internals. It contains four major components (the service interface with AAAC is not represented):

- A QoS Broker that performs control admission decisions, and overall network management tasks;
- A SPAAQE driver, that interfaces with SPAAQE entities from its network;
- An AQMUA driver, used to communicate with other AQMUA entities, in order to implement a distributed environment.
- A configuration and control graphical user interface.

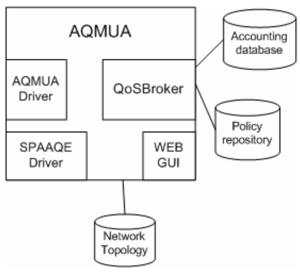

Fig. 2. AQMUA internals

In order to store all the information it needs, the AQMUA has three databases: an accounting database where it registers the network usage; a policy database, where it has service descriptions in terms of QoS parameters; and a network topology database, where it keeps the network element description and utilization, as well as the network architecture of neighbor areas. Furthermore, AQMUA can be configured using a Web GUI, where network elements can be listed into the database and their characteristics can be defined.

The QoSBroker is the key element in AQMUA. It is the QoSBroker that makes AQMUA resource management decisions. After a resource request, the QoSBroker examines the network resources availability, decides upon the acceptance of that traffic flow, and informs the SPAAQE. Information related with the authorized traffic flows is inserted in the accounting database, and can be used later for admission control decisions (deriving trends) or for accounting purposes.

SPAAQE

In this distributed architecture, the SPAAQE is the element positioned at the interface with the access network. The SPAAQE is responsible for detecting; classifying and processing data flow's going from and into the DiffServ network core. Typically this entity would act as a DiffServ Edge Router [3], but the added functionalities and individual processing capabilities implemented in SPAAQE lead us to a significantly different entity. In the SPAAQE the flow's are not only treated for QoS, but further processed in terms of signalling, accounting, destination address translation and any other kind of processing the Service Provider might want to add through the use of plug-ins. All decisions made by SPAAQE are local and relative to flow processing done by means of information received from the AQMUA: the SPAAQE does not take decisions by itself. However, once configured by the AQMUA, the SPAAQE is capable of handling localized decisions based on the information exchanged with the AQMUA.

One of the SPAAQE's major improvements is the capability to handle micro-flows identifiable by selectable mechanisms. These flows represent a greater granularity in the way the DiffServ network can differentiate handled traffic. For instance, they can represent specific User/Application traffic pairs that can provide the network administrator new ways of customizing available services to each specific client. This provides an IntServ-granularity applicable to user applications even when they are not IntServ-aware. Furthermore, these features are provided without any constrains to the final user. This is a consequence of SPAAQE main objective: a flexible signalling interface for the network provider, totally transparent to the end-user.

The SPAAQE's architecture (Fig. 3) consists of four main elements, a Classifier, a Scheduler, an Accounting module and an Advanced QoS Agent (ASQA). The Classifier is capable of classifying incoming packets based on information provided by the ASQA. The Scheduler manages queues, and is responsible for enforcing quality of service on packets going through SPAAQE; it further includes dropping capabilities (policing). The Accounting module interfaces with the accounting system deployed in the network (either Diameter or Radius servers) providing information on network usage. This module interfaces with the scheduler and ASQA through pluggable interfaces.

Fig. 3 also shows a simplified vision of the information flow inside the SPAAQE. Packets going through the Classifier are marked according to the installed configuration. Packets that don't match any configuration directive are considered unmarked and routed to the ASQA for further processing. The Classifier also routes packets that require advanced processing to the ASQA, such as packets it has no information on which action to take.

The ASQA is the SPAAQE's main element, where all control tasks are processed and treated. The ASQA is not only responsible for configuring the Classifier and

Scheduler but is the responsible for the advanced functionalities that differentiate the SPAAQE from a common Edge Router.

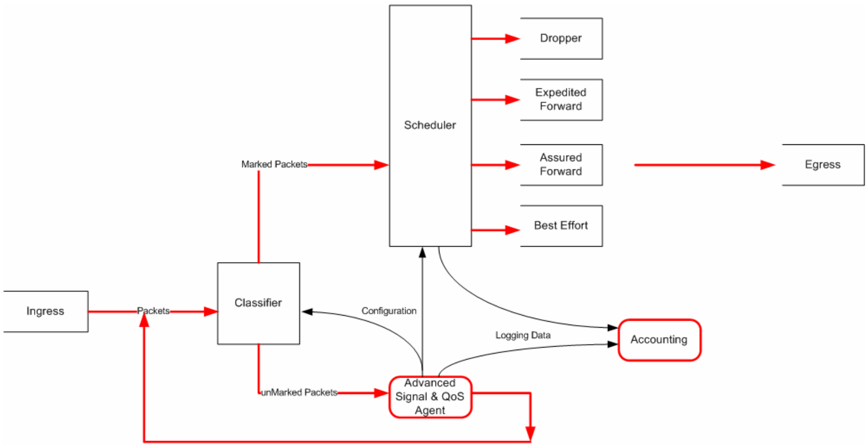

Fig. 3. SPAAQE Architecture and information flow

The ASQA is made of three functions; a Flow Detection unit capable of identifying micro-flows; a Flow Processing unit that treats the packet flows according to adequate configuration and a Flow Management process, which controls the SPAAQE. The ASQA is also the element that interfaces with the AQMUA, requesting instructions, providing information and receiving commands by means of COPS messages. Thus the AQMUA keeps a permanent knowledge of the current state of the network.

The most powerful feature of SPAAQE is the capability of the Flow Management process in the ASQA to process signalling messages, such as RSVP packets or SIP messages. This feature provides the means of achieving the desired network functionality for providing a homogeneous signalling to the core entities, while fully supporting any type of signalling in the access network. For instance, in terms of RSVP messages, the Flow Management receives the RSVP packets and according to information exchanged with the AQMUA enforces adequate QoS to the flow packets. The Flow Management process is not only capable of interpreting the RSVP packets and further translate IntServ into DiffServ, but is also capable of generating RSVP packets – which implies translating DiffServ-marked packets back into a IntServ flow - and issuing the proper messages to the applications.

This advanced feature is modular, in the sense that new modules capable of processing other signalling protocols can be incorporated into the SPAAQE, even during run-time, enabling it to translate any signalling protocol that the operator defines.

This *transsignalling* capability separates user-related signalling from network internal mechanisms. The transsignalling provided by the ASQA enables precise means of authorizing and enforcing QoS upon applications running on the operator Network. These features constitute a set of mechanisms that provide additional intelligence to the routers, enabling the deployment of advanced services (as per Application QoS) and support for legacy heterogeneous applications - while keeping a single signalling framework in the operator's core network. The classifier could also enable mechanisms of tracing and stopping Internet traffic; for instance, virus could be detected and stopped from spreading through the core network.

TransSignalling Usage

TransSignalling capabilities can potentially have multiple applications in future operator networks, especially when multiple heterogeneous environments are considered to be seamlessly supported – the usual network assumption for 4G scenarios [9].

Signaling heterogeneity

Handling signaling in a heterogeneous network can be a very complex task, as different frameworks have their own signaling mechanisms (e.g. the multiple QoS frameworks supported by IETF). Furthermore, several signaling schemes would optimally require proper integration with network support, in terms of QoS. For example, SIP messages should trigger also network level reservation [1]. This implies that QoS-related signaling has to be generated from non-QoS signaling, or for non-QoS aware applications.

When multiple types of clients or applications are connected to the same access point, the access router has to handle these diverse types of signaling. Pushing this intelligence to the Broker (AQMUA) would lead to low performance implementations. Furthermore, these signaling protocols would all need to understand how the network is internally managed. All these problems disappear using the SPAAQE. The interface between SPAAQE and the AQMUA is uniform, COPS-based. The applications have their own signaling mechanisms, and the SPAAQE has the required intelligence to hide this signaling from the network, translating the application messages (e.g. a SIP message) into QoS requests, and simplifying the overall network QoS management.

A similar situation happens with cellular wireless access networks (e.g. UMTS). These technologies have very specific QoS mechanisms and supporting QoS at the physical layer may require a complex set of messages exchanged between the mobile node and the access point/base station (which incorporates the access router in our model [6]). The SPAAQE is able to provide this physical layer signaling adaptation.

This system can be applied even for QoS unaware applications (such as common Internet gaming applications). The cost of changing all the network client applications is very high, so legacy applications have to be supported in future networks. In those cases, the SPAAQE can identify the application (or the user) and query AQMUA in order to know the QoS profile that should be used by this traffic. AQMUA answers can depend on the application that generated the traffic, allowing (for example) the operator to provide different QoS services to email and web browsing traffics.

Integration of different QoS architectures

Integration of different QoS architectures (namely IntServ and DiffServ) is a good illustrative example of this type of capabilities, and this module is already operational in our current SPAAQE implementation. This implementation is based in proven Linux API's such as Netfilter [10], TC [11] and L7 Filters [12]. These API's have been extended in functionalities, but maintaining their stability and scalability features as well as good performance. DiffServ approaches rely on DSCP marking associated to each packet, while IntServ relies on RSVP signaling. The integration of IntServ traffic (common on the access) and DiffServ networks (common on the operator network) has some problems, as these two frameworks use two different signaling strategies.

Our QoS system is able to decide the signal adaptation that should be made in order to provide the correct QoS signaling to the next network to be visited. The SPAAQE is a network stateful entity capable to change QoS signaling, in both directions. The AQMUA knows the network topology as well as the network QoS architecture. When traffic comes from a network border, SPAAQE queries AQMUA about the next network technology. In the case of the QoS architecture still being the same, SPAAQE does nothing, except routing the flows. But if the next network uses a different QoS framework, SPAAQE requests to the next network valid QoS profiles and "formats" the traffic with the correct QoS parameters. In a typical end-to-end communication, RSVP reservations are "logically" propagated along the AQMUA interoperation network, and restored at the end access link by the SPAAQE, while traffic is transmitted under a DiffServ QoS-framework.

Fig. 4 depicts this process. Upon the request for a new RSVP flow (1, a PATH message) the SPAAQE that lays on the Access Network requests its AQMUA for access authorization and associated QoS Profile for the requested flow. This request triggers an overall end-to-end resource availability evaluation. This AQMUA requests information on whether resources are available or not in the networks the flow must traverse by conducting several requests in chain to the AQMUAs managing the successive QoS domains (3, 4, 5, 6, COPS messages). As soon as each AQMUA gets a acknowledgement that the flow can go through, it issues a COPS-PR message to its border SPAAQE's (7, 8, 9). On the destination Access Network the SPAAQE hides the whole process from the application by engaging in an appropriate RSVP negotiation process with the destination (12 – PATH, and 13 - RESV).

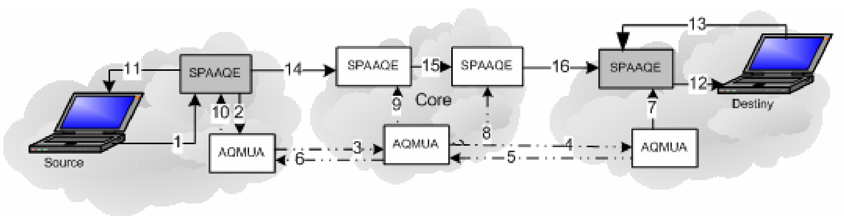

Fig. 4. Architecture Deployment example: Intserv-Diffserv interoperation

Finally the initial COPS Request (2) is answered (10) and the "RSVP" negotiation is completed (11 - RESV). When the communication flows through the network (14, 15, 16) the traffic presents the desired QoS performance, based on the previously provisioned QoS information, and an optimal mapping to the DiffServ network classes.

In this process, the AQMUA processing occupies a minimal percentage of time: path transversal is the dominant delay.

Conclusions

We have proposed a flexible heterogeneous QoS architecture that can be overlaid in any operator IP-based QoS network. This architecture can handle most problems created by QoS signaling heterogeneity and by traffic flowing between different QoS environments, through the interoperation between an AQMUA and complex SPAQQE units.

The proposed architecture enables an end-to-end QoS support independently of the QoS architectures used in the access networks. Traffic QoS requests are always "adapted" to the existing architecture in the next neighbor network, enabling the reutilization of the existing network elements and architectures. Furthermore, integration of network QoS signaling with non-QoS related signaling schemes (e.g. SIP) is also supported simply by adding a new processing module.

Current tests with IntServ - DiffServ signaling adaptation show that the transsignalling unit can process hundreds of flows per second in current hardware, for practical cases of RSVP applications.

References

1. L. Veltri, S. Salsano, D. Papalilo, “SIP Extensions for QoS support”, draft-veltri-sip-qsip-01, October 2002.
2. R.Braden, D.Clark, S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC 1633, July 1994.
3. S. Blake, D. Clarke *et all*, “An Architecture for Differentiated Services”, RFC 2475, December 1998.
4. M Gilmer; T. Braun, “Evaluation of bandwidth broker signalling”, ICNP '99, 7th Int. Conf. on Network Protocols, 1999.
5. Eunkyu Lee, Sang-Ick Byun, Myungchul Kim, "A Translator between Integrated Service/RSVP and Differentiated Service for End-to-End QoS", ICT 2003, 10th Int. Conf. on Telecommunications, Vol. 2, 2003.
6. Werner Almesberger, Silvia Giordano, *et all*, "A Prototype Implementation for IntServ Operation over DiffServ Networks", IEEE Globecom, S. Francisco, Dec. 2000.
7. H. Bai, M. Atiquzzaman, W. Ivancic, "Running Integrated Services over Differentiated Service Networks: Quantitative Performance Measurements", Quality of Service over Next-Generation Internet, Boston, Aug. 2002.
8. H. Einsiedler, Rui L. Aguiar, *et all*, “The Moby Dick Project: a Mobile Heterogeneous All-IP Architecture”, Advanced Technologies, Applications and Market Strategies for 3G - ATAMS 2001, ISBN 3-88309-20-X, pp. 164-171, 2001
9. Victor Marques, Rui L. Aguiar, *et all*, “An IP-based QoS architecture for 4G operator scenarios”, IEEE Wireless Communications Magazine, June 2003.
10. “Netfilter”, http://www.netfilter.org/
11. “Linux Advanced Routing & Traffic Control”, http://lartc.org/
12. “Application Layer Packet Classifier for Linux”, http://l7-filter.sourceforge.net/