A Quality of Service Management Architecture (QoSMA): A preliminary study

Marco Alfano
alfano@icsi.berkeley.edu

TR-95-070
December 1995

International Computer Science Institute
Berkeley, California

Abstract

The widespread use of distributed multimedia applications is posing new challenges in the management of resources for guaranteeing Quality of Service (QoS). For applications relying on the transfer of multimedia information, and in particular continuous media, it is essential that QoS is guaranteed at any level of the distributed system, including the operating system, the transport protocol, and the underlying network. Enhanced protocol support for end-to-end QoS negotiation, renegotiation, and indication of QoS degradation is also required. Little attention, however, has so far been paid to the definition of a coherent framework that incorporates QoS interfaces, management and mechanisms across all the layers of a management architecture. This paper describes a preliminary study in the development of an integrated Quality of Service Management Architecture (QoSMA) which offers a framework to specify and manage the required performance properties of multimedia applications over heterogeneous distributed systems.

1. Introduction

The spread of distributed multimedia applications is setting forth a new set of challenges in networking, including managing network resources for guaranteeing Quality of Service (QoS) [8], [20], [29], [30]. As users become more familiar with multimedia services, QoS must be also approached from the user point of view rather than the only network-oriented view that has been mainly addressed so far [4], [10], [13], [15]. The user must be given the possibility to express his requirements for the receiving service in terms of QoS parameters familiar to him and these parameters will be in turn translated into other parameters suitable for the distributed system and the underlying network [18].

Different performance requirements are addressed by different applications and even a single application often requires more than a QoS requirement [11]. For example, the high performance necessary for real-time multimedia services encompasses not only speed, but also reliability and availability, integrity, operability, delay, and accuracy, including synchronization accuracy between media streams.

Many advanced multimedia services are time-critical and need management support for ensuring agreed QoS [14], [28]. In this case, management has to promote QoS guarantees for each level of the system [25] because the overall QoS depends on the combined QoS of the distributed platform and underlying network [16]. The end-to-end management therefore has to include management capabilities for each layer participating to the service, i. e., from the service layer down to the network layer. As a consequence, the end-to-end management has to adopt a *vertical* architecture [12], [15].

In addition, multimedia services usually cross several networks and administrative domains [19], [21], [22], [26]. Management of such services therefore involves both security and inter-domain management issues, such as the problem of how several autonomous domains, both public and private, can cooperate to provide end-to-end QoS management and how network operators can make available the required functionality to service providers over the network/service management boundary. It also requires an understanding of the intra-domain network capabilities in order to integrate them with the end-to-end inter-domain capabilities. We have then a *horizontal* management architecture that spans across different network domains [15], [27]. In this paper we only focus on a vertical management architecture for QoS leaving the problems that arise when the network spans different domains for further studies.

2. Related work

Some studies have been done in order to create a comprehensive architectural framework for QoS, mainly with regards to the vertical approach. The International Standard Organization (ISO) has developed a set of standards for computer communication in the form of the seven-layer Reference Model for Open Systems Interconnection (OSI-RM), and these standards are now mature and popular. However, the OSI-RM evolved in an environment of data-only applications running over low-speed networks, and the QoS support provided by the OSI-RM reflects the limited QoS requirements of this class of applications. There is an ongoing joint project between ISO and the International Telecommunication Union (ITU) for developing a “Quality of Service Framework” [17] whose goal is to enable the future extensions of OSI standards in the direction of QoS provision by defining a reference architecture and standard terminology. It describes a set of concepts, services and mechanisms that can be applied to all OSI layers and to OSI management.

The key QoS framework concepts include:

- QoS characteristics;
- QoS requirements;
- QoS measures;
- QoS mechanisms;
- QoS management functions.

We will discuss these objects in more details in the next session because they represent, in our opinion, the basic objects that should appear in any management architecture for QoS provision.

The QoS framework, which is outlined in Fig. 1, is made up of two types of management entities that attempt to meet the QoS requirements by monitoring, maintaining and controlling the QoS parameters (requirements and measurements):

- *Layer QoS entities*, entities associated within the operation of a particular (N)-subsystem. They implement direct control of protocol entities, etc., that are necessary for support of the QoS requirements made by the system.
- *System QoS entities*, entities which have a system wide role. They interact with layer QoS entities to monitor and control the performance of the system. In addition, they may implement managed objects as means by which systems management entities may interact with the provision of QoS in the system.

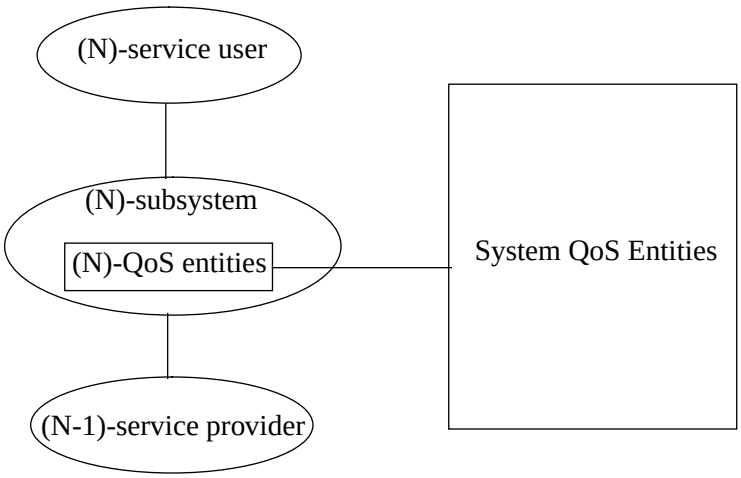

Fig. 1. QoS framework model.

Another study has been carried out at the Lancaster University [8], [16] and a QoS architecture (QOS-A) has been proposed. It is a layered architecture of services and mechanisms for QoS management in an environment based on ISO's Open Distributed Processing (ODP) standards and ATM networks. This architecture is aimed to the support of continuous media applications and is vertically organized in five layers as depicted in Fig. 2. The upper layer consists of a *distributed applications platform* provided by an ODP-compatible distributed systems platform. Supporting

this is a *transport layer* which contains a range of QoS-configurable protocols. For example, separate protocols are provided for continuous media and constrained latency message QoS.

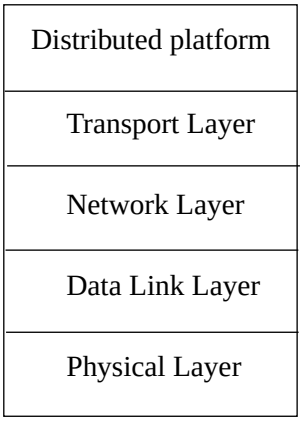

Fig. 2. Lancaster QOS-A architecture.

A significant work on QoS provision at the transport and network levels comes from the Tenet Group at the University of California at Berkeley. This group has developed a family of real-time protocols [6], [7], [24] that run in parallel to TCP/IP (Fig. 3).

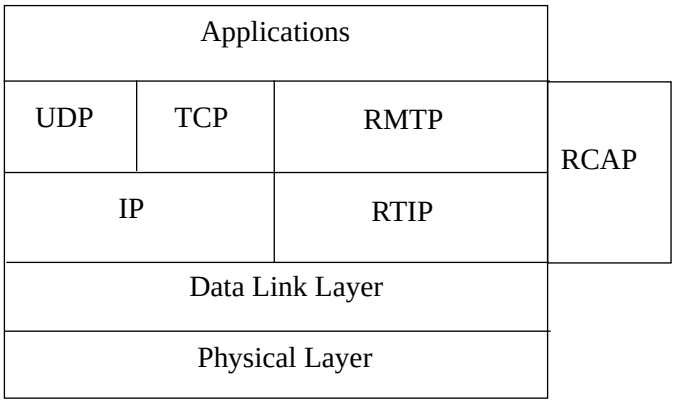

Fig. 3. Tenet protocol stack and the corresponding Internet protocols.

The protocol family consists of the “Real-Time Internet Protocol” (RTIP) which guarantees end-to-end packet delivery at the network layer, the “Real-time Message Transport Protocol” (RMTP) that sits above RTIP and provides sequenced and periodic delivery of messages with QoS control over throughput, delays, and error bounds, and “Real-time Channel Administration Protocol” (RCAP) that is responsible for channel establishment, status reporting and channel tear down.

Another important work at the low levels of the management architecture has been carried out at the Columbia University [9], [23]. The main goal is the allocation and control of network resources under QoS constraints. This is pursued through a traffic control system for multimedia

networks that contains a collection of resource control subsystems, each implementing a specific task such as admission control or routing. Each of these subsystems regulates access to a specific resource, by responding to requests that are generated by functions external to the resource control subsystem.

The present work proposes the basis of a QoS management architecture (QoSMA) that spans from the network layer up to the application layer and discusses the interactions of external entities (user, network provider, and service provider) with the different layers. The paper is organized as follows. Section 2 presents the main QoS objects of the ISO architecture. Section 3 presents the proposed QoS management architecture. Finally, Section 5 presents some conclusions and future work.

3. QoS objects

As discussed above, the QoS Framework [17] that is being developed as ISO and ITU standards contains different objects. The main objects are:

- *QoS characteristic*, a quantifiable aspect of QoS, which is defined independently of the means by which it is represented or controlled;
- *QoS requirement*, that determines what QoS level is requested;
- *QoS measure*, that supplies one or more observed values relating to a QoS characteristic;
- *QoS mechanisms*, that provide support establishment, monitoring, maintenance, control, or enquiry of QoS;
- *QoS management function*, that is a function designed to meet QoS requirements by means of one or more QoS mechanisms.

We believe that these objects should be part of any management architecture whose goal is to provide a service with required QoS to the end user. We now examine these objects in more details.

3.1 QoS characteristics

As said above, a QoS characteristic is a quantifiable aspect of QoS. It represents the true underlying state of affairs, as opposed to any measurement or control parameter and can therefore be thought as a quantity in a mathematical model of a system. QoS characteristics are intended to be used to model the actual behaviour of systems.

Many QoS characteristics can be applied to a variety of circumstances. The basic idea is to define a *generic characteristic* independently of what it is applied to and then define various *derivations* that may or must be applied in order to make the characteristic concrete and usable in practice. So, for example, if *time delay* is considered to be a generic characteristic, a derived characteristic will be *transit delay* and another will be *request/reply delay.*

The QoS generic characteristics of importance to OSI are grouped as follows:

Time related characteristics

- date/time;
- time delay;

- coherence;
- data time validity.

Capacity related characteristics

- capacity;
- throughput.

Integrity related characteristics

- accuracy;
- safety.

Cost related characteristics

- cost.

Security related characteristics

- protection.

Reliability related characteristics

- availability;
- reliability.

3.2 QoS requirements

As said above, QoS requirements refer to what is requested on QoS characteristics. Requirements can be expressed in many different ways:

- a desired level of characteristic, e.g., a target of some kind;
- a maximum or minimum level of a characteristic, e.g., a bound;
- a measured value, used to convey historical information;
- a threshold level;
- a warning or signal to take corrective action.

QoS requirements may relate to a number of QoS characteristics and, at least in principle, may express trade-offs between them. QoS requirements may apply to one or more information transfers or interactions, e.g., over a given period of time.

3.3 QoS measures

QoS measures are used during service provisioning to indicate one or more values relating to QoS characteristics. A characteristic is then monitored by means of its measure and if the obtained value does not correspond to the desired requirement, corrective actions are taken.

3.4 QoS mechanisms and QoS management functions

A number of functions are used to manage QoS in order to meet QoS requirements. The term *QoS management function* is used to identify such functions. QoS management functions may require different types of action to be performed: negotiation, admission control and monitoring, for example. It is therefore useful to regard them as composed of a number of smaller elements, termed *QoS mechanisms*, which can be specified independently. QoS management functions are then described as being performed by one or more QoS mechanisms operating in sequence or in parallel.

QoS mechanisms perform or support a number of activities related to QoS, namely:

- QoS establishment;
- QoS monitoring;
- QoS alert;
- QoS maintenance;
- QoS control;
- QoS enquiry.

4. A QoS management architecture

In this section we describe the basis of a QoS management architecture (QoSMA) which follows the vertical approach and uses the QoS objects (that must be particularized for each layer of the architecture) described in the previous section. This architecture is made up of four layers and is depicted in Fig. 4. There are three external entities, namely *user*, *service provider*, and *network provider* that represent the characters that more likely play a basic role in the deployment of a multimedia service. The arrows from one entity to an architecture layer means that the entity may interact with that layer. Solid arrows indicate which interactions are, in our opinion, more likely to be allowed. Dotted arrows indicate interactions that may or may not exist, depending on the context.

Users are more likely to interact only with the service layer. The interaction with the underlying layers may be allowed, for example, if the user is the programmer of the application and has access (even partially) to the system resources. In Fig. 4 we are assuming that the service provider is managing the computing resources of the system so he is the only one to access the operating system and distributed platform sub-layers. If this is not the case, a new entity, a *distributed system provider* (that can correspond to the network provider) should be introduced. The network provider is the one who interacts with the lower layers of the architecture, i.e., the layers that deal with the communication part of multimedia services. We now examine the various layers of the architecture.

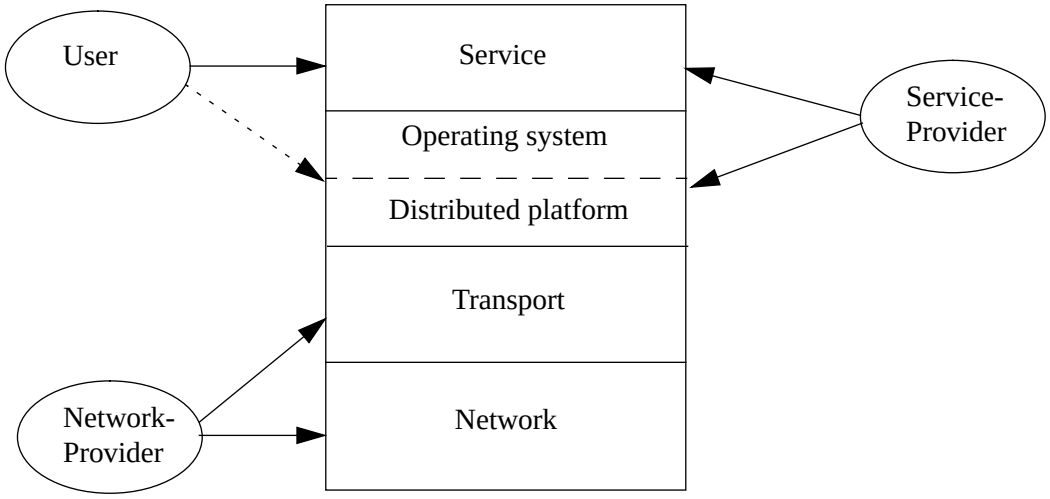

Fig. 4. Proposed QoS management architecture and external interactions.

Service

The *service* (application) layer is the one which the user will more likely interact with. QoS specifications at this level should be meaningful to the user and lower-level considerations such as the rate and burst size of a transport connection should be hidden. Moreover it is important that QoS specifications are qualitative (e.g., high-quality or low-quality video) rather than quantitative. A mechanism must then be provided in order to translate user requirements into quantitative QoS requirements that are meaningful to the lower layers.

Distributed platform and operating system

The next layer is a combination of two sub-layers: *operating system* and *distributed platform*. This is done to resemble the way most distributed systems are presently built, i.e., by linking machines with their own operating systems (e.g., Unix) and adding to each machine the suitable software for communicating (and then co-operating) with other machines. Operating system and distributed platform sub-layers will receive QoS requirements from the service layer and will execute the suitable QoS mechanisms to satisfy those requirements.

In order to support multimedia services, operating systems must provide a degree of QoS support to uphold the real time isochronous nature of continuous media data such as audio and video. Performance of the scheduling mechanism of the operating system will be monitored to ensure that timely data arriving from the network are also delivered in time to their final destination. QoS driven operating systems and distributed platforms are then required to support existing applications and simultaneously offer predictable performance in a dynamic and unpredictable environment [2], [3], [5]. Resource management strategies are required for all areas of operating system management including processor scheduling, device management, and memory management.

Transport

The *transport* layer has to provide support for different kinds of transfer services such as continuous media, transaction-oriented, and data transfer services and these services should be able to operate both as connection-oriented and connectionless. Moreover, this layer should provide for multicast connections and synchronization between multimedia streams (e.g., audio and video).

The transport layer receives the QoS requirements regarding the communication part of the multimedia service from the distributed platform sub-layer and applies the suitable QoS mechanisms to meet those requirements.

Network

The *network* layer works with the transport layer to meet the QoS requirements on communication. Usually a small number of fixed levels is used to specify the commitment to the allowed traffic in the network:

- *deterministic* commitment, which is typically used to guarantee QoS for hard real time performance applications;
- *statistical* commitment, which allows for a certain percentage of violations in the requested QoS and is particularly suitable for continuous media applications;
- *best effort* commitment, the lowest priority commitment and synonymous with a datagram service. No network resources are allocated or monitored because the network provider does not have to guarantee any level of service. This commitment level only receives whatever network resources are available after the other levels have been served.

5. Conclusions and future work

There are many important issues that require further research work. As already pointed out, work on QoS has concentrated on the network and communication infrastructure. Recently, however, the topic has become increasingly important in an end-system context because of the interest in operating system support for multimedia applications. It is becoming recognized in the multimedia community that classes of applications exist which must actively manipulate real-time continuous media data in an operating system environment. QoS must then be considered as an end-to-end issue, i.e., from application to application. This requires careful coordination of disk scheduling, process scheduling and the various layers of communication protocols. For example, to implement an audio connection with a given QoS, it is necessary to achieve the desired QoS in the transport protocol and to schedule processes at the desired rate with the arrival of audio data. Such integration should eventually extend to areas such as device and memory management. Overall, it is necessary for applications with varying needs and assumptions to coexist in an operating system environment able to simultaneously satisfy all their various requirements. There is then a need to have real systems that are able to effectively run batch mode, interactive, and continuous media applications at the same time.

Another important issue that needs further research is related to QoS characteristics and requirements on them. QoS characteristics are not completely unrelated but often a characteristic influences one or more other characteristics. For example, time-related characteristics can be somehow incompatible with reliability-related characteristics if the latter are obtained by means of process or message replication. In this case an increase of reliability will entail a possible overload of the

system and in turn greater time delays. Thus, for this particular case a trade-off among the two characteristics should be considered.

Summarizing, some of the most important issues in QoS management that in our opinion require further research work are the following:

- QoS must be regarded as an end-to-end issue involving the application, the distributed platform, and the communication infrastructure. Co-ordination among all these elements is necessary so that a service is provided with the desired QoS established by the user and the service provider. Scheduling of processes in a distributed platform [1] and scheduling of the communication channel [20], [29] play a basic role and must be somehow co-ordinated.
- A mechanism that precisely maps QoS characteristics and requirements from an upper layer to a lower layer is required. A user must be granted the possibility to express his requirements in terms of attributes (mainly in a qualitative way) that are familiar to him and related to his view of the service. These requirements then need to be precisely mapped in QoS requirements for the lower layers (distributed system and communication infrastructure).
- Characteristics sometimes influence each other. When specifying requirements on more than one characteristic, one must be sure that all of the requirements can be satisfied and whether the satisfaction of a requirement entails that other characteristics cannot reach the desired value. Sometimes, a trade-off among different characteristics must be considered.

Acknowledgments

This work has partly been developed under a fellowship of the Italian National Council of Research (CNR) at the Dipartimento di Informatica e Sistemistica, Universita' "La Sapienza" in Rome.

References

[1] M. Alfano. Scheduling features in distributed systems. *Proc. of the SBT/IEEE International Telecommunications Symposium*, Rio de Janeiro, August 1994, pp. 52-56.

[2] M. Alfano et al. Scheduling distributed algorithms on heterogeneous computer networks. *Application of High-Performance Computing in Engineering IV*. Ed. H. Power. Computational Mechanics Publications, 1995. pp. 47-54.

[3] M. Alfano. Uno scheduler per applicazioni distribuite guidato da specifiche di utente. *Ph.D thesis, University of Palermo*, February 1995.

[4] B. Alpers. Applying domain and policy concepts to customer network management. *Proc. ISS `95*. Berlin, April 1995, Vol. 2, pp. 356-360.

[5] D.P. Anderson. Metascheduling for continuous media. *ACM Transactions on Computer Systems.* Vol. 11, No. 3, August 1993, pp. 226-252.

[6] A. Banerjea et al. The Tenet real-time protocol suite: Design, implementation and experiences. *International Computer Science Institute Technical Report TR-94-059*, November 1994.

[7] A. Banerjea et al. Experiments with the Tenet real-time protocol suite on the Sequoia 2000 wide area network. *Proc ACM Multimedia '94*, San Francisco, October 1994, pp. 183-192.

[8] A. Campbell et al. Integrated Quality of Service for multimedia communications. *Proc. IEEE INFOCOM '93,* San Francisco, March 1993, pp. 732-739.

[9] M.C. Chan et al. Managing real-time services in multimedia networks using dynamic visualization and high-level controls. *Proc. ACM Multimedia '95,* San Francisco, November 1995, pp. 243-253.

[10] S. Cronjaeger and P. Vindeby. Translation of multimedia user requirements into ATM switching requirements by using appropriate transport protocols. *Proc. ISS `95.* Berlin, April 1995, Vol. 1, pp. 180-184.

[11] D. Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine,* November 1990, pp. 65-72

[12] F. Fischer et al. Layered network management within the D1-Network. *Proc. ISS '95 Berlin,* April 1995, Vol. 2, pp. 435-439

[13] B. Gadher et al. A distributed broadband metropolitan network for residential multimedia applications. *Proc. ISS `95,* Berlin, April 1995, Vol. 2, pp. 190-194

[14] F. Georges. Performance evaluation & management of interconnected networks. *Proc. SBT/ IEEE Int. Telecommun. Symposium.* Rio de Janeiro, August 1994, pp. 322-326

[15] J. Hall et al. Customer requirements on telesrvice management. *Integrated Network Management IV,* Ed. A.S. Sethi et al. Chapman & Hall 1995, pp. 143-155

[16] D. Hutchison et al. Quality of service management in distributed systems. *Network and Distributed Systems Management.* Ed. M. Sloman Addison-Wesley 1994, ch. 11, pp. 273-302

[17] ISO/IEC. Quality of Service - Basic Framework. *ISO/IEC JTC1/SC2/WG1 N8871,* August 1994

[18] J. Jung and D. Seret. Translation of QoS parameters into ATM performance parameters in B-ISDN *Proc. IEEE INFOCOM `93,* San Francisco, March 1993, pp. 748-755

[19] I. Katzela et al. Centralized vs distributed fault localization. *Integrated Network Management IV,* Ed. A.S. Sethi et al. Chapman & Hall 1995, pp. 250-261 5

[20] A.A. Lazar and G. Pacifici. Control of resources in broadband networks with Quality of Service guarantees. *IEEE Communications Magazine,* October 1991, pp. 66-73

[21] T. Magedanz et al. Towards Pan-European IN services: IN interworking and IN management issues. *Proc. ISS `95 Berlin,* April 1995, Vol. 2, pp. 112-116

[22] K. Motomura et al. Management integration technologies. *NTT Review,* Vol. 7, No. 2, March 1995, pp. 66-74

[23] G. Pacifici and R. Stadler. An architecture for performance management of multimedia networks. *Integrated Network Management IV,* Ed. A.S. Sethi et al. Chapman & Hall 1995, pp. 174-186

[24] C. Parris et al. A dynamic management scheme for real-time connections. *Proc. IEEE INFOCOM '94,* Toronto, June 1994, pp. 698-707.

[25] K.J. Putz Field trials of multimedia applications in ATM networks: Quality-of-Service, functional and implementation aspects. *Proc. ISS `95,* Berlin, April 1995, Vol. 2, pp. 180-184

[26] S. Rabie. Traffic management in multi-service enterprise networks. *Proc. ISS `95,* Berlin, April 1995, Vol. 1, pp. 336-340

[27] P. Ray and M. Fry. Integrated service management within heterogeneous environments. *Proc. SBT/IEEE Int. Telecommun. Symposium,* Rio de Janeiro, August 1994, pp. 347-351

[28] F. Somers and M. Edholm. Intelligent resource dimensioning in ATM networks *Proc. ISS `95,* Berlin, April 1995, Vol. 2, pp. 62-66

[29] M. Woo et al. A synchronization framework for communication of pre-orchestrated multimedia information. *IEEE Network,* January/February 1994, pp. 52-61

[30] N. Yamanaka et al. Full-Net: A flexible multi-QoS ATM network based on a logically configured VC-network. *Proc. ISS `95* Berlin, April 1995, Vol. 2, pp. 195-199.