Quality of Service in Communication Systems – Challenges and Approaches

Authors:
Gonçalo Quadros, Edmundo Monteiro, Fernando Boavida

Affiliation and Address:
Universidade de Coimbra, Departamento de Engenharia Informática,
Pólo II - Pinhal de Marrocos, 3030 Coimbra, PORTUGAL
e-mail: {quadros, edmundo, boavida}@dei.uc.pt, Tel: +351 39 790000, Fax: +351 39 701266

Abstract

Several approaches have been proposed to empower communication systems with quality of service (QoS) capabilities. In general, their main objective is to define ways to coherently support end-to-end performance application needs, based on the establishment of and agreement on a set of concepts, policies and mechanisms.

The first part of this paper has a tutorial nature, and analyses a representative part of the work that has been produced in this field by the computer communications and standards communities, namely the ISO/IEC QoS framework, the IETF Integrated Services model, the QoS Architecture, the ATM traffic and congestion control model and the Tenet model. The main purpose of this analysis is to capture the nature of the issues associated with the implementation of communication systems able to provide end-to-end QoS to their users. The main conclusion of this analysis is that an important challenge associated with the development of QoS-capable communication systems is the conception of an effective QoS monitoring function supported by a QoS metric that is able to provide significant information on the communication system performance.

In the second part of this paper we propose a way to measure the quality of service provide by communication systems. The proposed metric was implemented, in order to test its basic concepts, to assess its feasibility and to measure the associated overhead. The results show that it is possible to measure the quality of service provided by communication systems with a reduced overhead and in a way that provides comparable QoS measurements, despite the diverse nature of the QoS characteristics associated with communication flows.

Keywords: QoS monitoring, packet switching communication systems.

1. Introduction

The quality of service concept is at the basis of an intense work in the computer and communication investigation arena. Dealing with quality of service is to deal with a large variety of issues and factors related to the performance of technologies used in information systems and to their impact on users.

Interactivity with human users imposes limits on information transport delay. Isochronous or continuous media traffic produced by multimedia applications imposes limits on *jitter*. The bursty nature of compressed video traffic requires available bandwidth in the communication system. Conversely, common nowadays applications, for example e-mail or file transfer, have strict reliability requirements.

The heterogeneous traffic produced by all these distinct applications will naturally use the same communication system, which has to behave differently according to the nature of carried data. Basically, applications expect different performances in terms of transport delay, bandwidth, reliability, failure recover, or session establishment latency [Partridge93].

Different approaches can be used to construct QoS-capable systems. One would be to over dimension the system, using the requirements put by worst load conditions, and assuming the resulting waste of resources. Another approach would be to use scale and filter mechanisms to adapt the generated load to the availability of resources, assuming that applications are prepared for adaptation and for the possibility of not always having enough QoS [Steinmetz97]. We believe, like R. Steinmetz and L. Wolf, that distributed computer systems will be for a long time in the "window of scarcity". So, to provide adequate QoS to applications, resource reservation and scheduling techniques must be used.

Given the above, in order to achieve communication systems with appropriate characteristics able to provide QoS to their users, several problems must be solved. First of all, it is necessary to provide a way for the applications to specify the characteristics of the traffic they will generate and their QoS requirements. In turn, communication systems must be able to verify if there are conditions to provide the QoS that is being asked for. This process must take into account QoS compromises already assumed by the system – the acceptance of a new *flow*[1] shouldn't degrade the QoS of existing ones. Thus, system performance control becomes an *on-line activity* [Kurose93].

[1] The notion of flow is a fundamental one in the QoS domain. It is usually defined as the production, transmission and, eventually, consumption of a sequence of data resulting from a single activity and regulated by a given quality of service specification [Campbell94].

After the acceptance of a new flow, the communication system must create conditions for its convenient support. It must reserve and allocate the necessary resources, and configure all the adjustable mechanisms related to QoS provision. Afterwards, those resources and mechanisms should be continuously monitored and managed in order to keep the system performance at reasonable levels.

Notice that for resource reservation it is necessary to establish a data path through the communication system. This path can be installed using *connection oriented* (*hard state*) technologies (as in ATM), or *soft state* technologies (as in RSVP, used with IP). In the last case the path establishment and resource reservation are executed at different times; the path and the resources allocated to a flow have a temporary nature and need to be continuously refreshed.

Different approaches - hereafter called models - have been proposed to empower communication systems with QoS capabilities. Few have a broad scope, that is an end-to-end perspective, considering also end system components (like CPU, memory or disks). In this paper we will analyse the following QoS models:

- The ISO/IEC QoS framework, which establishes a terminological reference and structures a set of concepts useful to construct QoS capable communication systems. It intents to provide a way to consistently develop and refine standards related to QoS [ISO/IEC9680];

- The Integrated Services (IS) model, proposed by the IETF ISWG[2], to turn possible the use of the Internet for multimedia applications support. It defines a set of services [Shenker97b, Wroclaw97] to be installed in network elements (e.g., routers, subnetworks and end-systems). The IS basic model is presented in [Braden94] and [Shenker97];

- The QoS Architecture (QoS-A), which is an architectural model, developed and constructed at Lancaster University [Campbell94]. It is one of the first models that considered not only QoS aspects related to communication systems but also QoS aspects related to end-system components.

- The ATM Traffic Control and Congestion Control model, that identifies a set of functions, procedures, parameters and descriptors, used to efficiently provide quality of service to ATM users [atmf-uni3.1].

---

[2] *Integrated Services Working Group* of the *Internet Engineering Task Force*. This work is complementary to the one being developed in RSVP WG for the definition of a resource reservation setup protocol to be used in the establishment of flows.

- The Tenet group[3] work, which established a set of principles to guide the implementation of real time communication systems – *Tenet approach* – along with a set of *Tenet schemes* and *Tenet suites* for the implementation of QoS capable communication systems [Ferrari94, Banerjea94, Gupta93].

Before we continue, we would like to mention a very recent approach to QoS control in the Internet - the *Differentiated Services* approach [DiffServ] - which is being discussed also in IETF. Using Jon Crowcroft words, "differentiated services refers to a thin/lightweight approach to providing some improvement to the Internet *Best Effort* service: (1) without defining explicit performance parameters, (2) without defining explicit parameters to be exchanged through signalling". Basically, in this approach, several bits are used in each IP packet, which allow the definition of several different classes. To each class is then given a different QoS level, basically in terms of delay and packet losses. Researchers interested in differentiated services are divided between those who think that integrated services and RSVP are useless and those who think that the two approaches are complementary. We include ourselves in the last group.

Section 2 of the present paper provides an analysis and comparison of the above mentioned QoS models. Section 3 generally presents a QoS metric to be applied in packet switched communication systems. Section 4 refers current work being developed by the authors.

2. QoS models analysis and comparison

As stated in [Steinmetz97], the operation of a QoS-capable communication system supported by resource reservation and scheduling, can grossly be divided in two steps (or phases): *QoS negotiation* and *QoS transmission*.

One fundamental aspect of any QoS-capable communication system, related to the negotiation phase, is the interface used to specify quality of service which, in turn, is related to the richness of the *service contract*[4]. In this section we will analyse this aspect for the different models under consideration. We will also analyse and compare some fundamental communication system characteristics, related to the transmission phase, which can help the reader to capture additional key aspects connected to QoS provision and the differences between the various QoS models.

---

[3] Formed in 1989, at the University of Berkley California and at the International Computer Science Institute.

[4] The final set of values related to the characteristics of the traffic to be transported and its QoS requirements.

2.1 Traffic characteristics and QoS needs specification

Table 1 summarises the traffic characteristics and QoS requirements that can be specified through the interface with the communication system, for each of the considered QoS models. All the interfaces are located at the transport layer, with the obvious exception of the one related to ATM, which is located at link or network layer.

To better understand the information presented in the referred table, some notes must be made. Traffic characteristics and QoS requirements used in IS the model concern the services already defined by IETF ISWG; other services may be defined in the future. The ISO/IEC framework is a conceptual proposal, thus, grossly, any quantifiable QoS characteristic can be specified (the model only tries to rule how this should be done). The Tenet approach provides two transport models with different associated QoS specifications: RMTP and CMTP; CMTP was designed to take advantage of particular characteristics of continuous media traffic.

As it can be seen in Table 1, traffic and QoS needs characterisation is commonly based on the specification of throughput, delay, jitter and loss. Throughput specification characterises traffic in a way that is essentially the same in all models. Jitter, delay, and loss specifications characterise the needed QoS. The former is not always used. The specifications of delay and losses are made using different methods.

One important aspect is that in some models the specification of QoS requirements is grouped in *classes*. In ATM UNI 3.1, for example, delay, jitter and losses are specified through a predefined class. This can also be the case for delay specification in IS, where losses are implicitly associated with the chosen service[5]. Making the characterisation of QoS requirements using classes can result in substantial reduction of systems complexity. Conversely, it can also result in an unnecessary limited system, given that we are not able to freely specify QoS needs. There’s no consensus on the better strategy to adopt. Nevertheless, notice that in ATM UNI 4.0 the specification of QoS requirements based on classes is not mandatory and it is supported only for compatibility reasons.

Notice also that all models specify MTU and mTU (implicitly or explicitly). This is needed because the operation of many low-level mechanisms (such as buffer or timer managers) depends on packet sizes.

[5] Depending on the service, applications will get *no queuing losses* or *litle queueing losses* in the network.

2.2 Other specifications characteristics

In certain models it is possible to make other specifications related to QoS, in addition to the ones made for traffic and QoS characterisation. For instance, QoS-A allows the specification of the desired communication system behaviour related to QoS monitoring and management, or even the mechanisms to be used in the service contract establishment – negotiation, anticipation or fast establishment.

The establishment of service contracts through negotiation enables a better match between the application needs and the resources available in communication systems. The problem is the complexity and the time that can be wasted on the negotiation process. The possibility to specify a fast contract establishment in QOS-A intents to overcome the time problem, allowing the transmission of initial data before the service contract is established. The Tenet group has assumed that the specification of QoS limits (maximum and minimum) as opposed to the specification of single target values, eliminates the need for iterative negotiations. ISO/IEC stipulates that QoS specifications, in fact, should be made using ranges, but does not preclude the use of negotiations for service contract establishment.

The specification of traffic characteristics and QoS requirements has, essentially, a quantitative nature. Given the diversity of applications, it is interesting to qualitatively refine this specification. For example, to determine that communication systems must give hard or deterministic QoS guarantees or, on the opposite extreme, only a *best-effort* try. The type of service (TS) is used for this purpose. Examples of type of service are: *deterministic guaranteed*, *statistically guaranteed*, *predictive*, *adaptive* or *best effort*. Some models allow TS to be specified in a way that is orthogonal to QoS characteristics, that is, it is possible to define different TS for each specified characteristic. To increase even more the model flexibility (but also the associated complexity) some models allow statistic, or probabilistic, specification of QoS characteristics (in the Tenet case, this is only possible for delay and loss).

A better definition of the communication system behaviour is also possible through the specification of actions to be triggered when certain conditions hold, for instance when minimum QoS limits are met. The ISO/IEC framework considers the use of pre-defined actions to be triggered off not only when QoS limits are attained but also when thresholds – which can also be specified– are surpassed.

For several types of applications, the possibility to anticipate the establishment of the service contract (to establish the service contract before the application becomes in fact active) is very interesting. Some of the presented models make proposals in this fundamental field.

Table 2 summarises the characteristics of the different models under consideration in respect to the previous discussion.

2.3 General communication systems characteristics

The organisation of the information to be transported in flows is fundamental in QoS-capable communication systems based on resource reservation and scheduling. Every model here presented considers flows that are simplex and can be multicast. Multimedia distributed applications demand the possibility to send information from one to many parties. So, in order to efficiently use communications resources a multicast capability is fundamental. The IS model goes a little further, by explicitly considering the use of many-to-many multicast technology (IP multicast). To avoid the wasting of resources, it is advantageous that different branches of multicast flows can receive different QoS (given that different end-systems can have different quality needs). Normally, this implies special negotiation mechanisms for flow establishment and depends on certain technological features: if, for example, the QoS request is *receiver oriented*[6] the provision of non uniform QoS in multicast connections becomes easier.

Again, to prevent the wasting of resources, as QoS needs can vary during the applications lifetime, it should be possible to dynamically change the established service contract. From all the models here presented, all but ATM provide QoS dynamically.

As traffic generated by certain applications should have higher precedence over general traffic it is interesting to construct mechanisms that allow flow prioritisation. In fact, despite the eventual absence of resources in a communication system at a given instant of time, it seems important to have mechanisms that avoid the rejection of urgent flows. Nevertheless, this subject hasn’t received too much research attention until now. In IS, albeit the absence of any reference to flow prioritisation, protocols used to establish flows do address this issue – the concept of *connection preemption* is present at least in some of them.

---

[6] Specified by the receiver end.

Table 3 summarises the comparison of models in what respects the topics just referred. It also sums up information related to two important issues for any communication system: pricing and security. It is evident that something must dissuade communication system users from always requesting top QoS. This is one of the roles of a well designed pricing policy. It is also obvious the need for security mechanisms. Only Tenet and QoS-A models deal with pricing issues, giving users the possibility to specify the maximum price they want to pay for the support of their flows. On the other hand, only the ISO/IEC framework and the Tenet model consider security aspects. Tenet researchers are actively developing security extensions to empower their model with authentication, authorisation, data confidentially and integrity services. Despite the fact that IS services do not address security, yet, it is important to note that there is some work in this domain made by the IETF, for instance in the context of RSVP.

In order to provide QoS to flows, different communication system functions or mechanisms must be used. First of all, it is necessary to define a flow path (ideally using a *QoS aware routing protocol*). In parallel, it is necessary to perform *admission control* to assess if the flow can be supported and *resource reservation* to create the necessary conditions for its support. As different technologies and communication layers are involved in the provision of QoS, it is necessary to perform *QoS mapping* (that is, to translate the quality of service specifications into something meaningful to each layer).

It is also necessary to continually configure low-level mechanisms related to the transport of data. For example, mechanisms for managing the forwarding of data (*flow scheduling*), mechanisms to control whether or not the traffic is being generated according to the service contract (*flow policing*), mechanisms to regulate flows in order to give them the expected characteristics (*flow shaping*), or mechanisms for flow control.

Lastly, it is fundamental to perform *QoS monitoring* and *QoS management*, in order to maintain the quality of service level within the contracted values. In [Aurreco95] different QoS models, including some of the ones presented here, are surveyed and compared based on the above mentioned mechanisms.

QoS monitoring is the function used to track the quality of service actually given to flows. The information it collects should provide a comprehensive, consistent and sufficiently detailed view of the quality of service provided by communication systems, which is, undoubtedly, important to manage and tune the mechanisms related to QoS support. Nevertheless, given the diverse nature of the objects to measure, it is difficult to obtain meaningful and comparable QoS state information on the communication system operation. This is an area with a clear lack of research effort, as it is proved by the absence of proposals for QoS monitoring.

2.4 Conclusion

In this section we have discussed several aspects related to the characteristics that QoS-capable systems should have. It seems clear that one of the biggest challenges related to their construction is the determination of the better trade-off between complexity, flexibility and functionality. For this determination, clear answers are needed for questions like:

Should we negotiate the service contract or not? Should we specify single QoS values or ranges? Is it really important to orthogonally specify TS, or to establish the service contract in advance? It is worthwhile to allow the specification of actions to be triggered when QoS degradation happens, or to allow the modification of the service contract during flows lifetime?

From the analysis made we notice that different approaches have been proposed, and tried, which essentially differ in the way they assume the mentioned trade-off. We noticed also a fundamental point: the lack of proposals to consistently measure QoS. The importance of this fact steams from our believe that the very first step to construct efficient and skilled QoS-capable systems should be the conception of an effective way to measure QoS.

Such a measurement mechanism will certainly facilitate the control and management of the provided QoS, enabling the construction of better systems, more efficient, with higher functionality and lower complexity. Even when different paradigms to construct QoS-capable systems are used − for instance *differentiated services* or systems based on QoS adaptation − the availability of a way to effectively measure QoS is undoubtedly very interesting.

Given the above statements, we propose a way to measure QoS, which is nuclear to the operation of the model we are also developing – the UC-QoS model [Monteiro95a, Monteiro95b]. A major goal of our proposal, from the very first stage of its design, was the development of a way to get normalised, meaningful, comprehensive and comparable QoS measures. In the following section, the referred metric is further discussed.

3. Measuring QoS in Packet Switched Communication Systems

This section generally presents a QoS metric to be applied in packet switched communication systems, which is detailed in [Quadros98]. It was targeted to measure quantifiable QoS characteristics [ISO/IEC9680], as throughput, delay, loss or jitter, in transport, network or lower layers. Despite that, the basic metric concepts can be easily applied to other systems where the need to measure the provided QoS does exist (as is the case, for instance, of operating systems or communication system higher layers). Therefore, the metric can be applied to comprehensive QoS architectures, providing an integrated and consistent way to measure, and to manage, QoS in networks and end-systems.

Different flows supported by a communication system will have different QoS needs. For instance, some will need very low losses, others low transit delay; some will have high tolerance to QoS degradation, others not so high. It is evident that QoS measures produced by some monitoring mechanism must consider this diversity or, otherwise, they will be useless.

The natural way to measure QoS is to quantify absolute values or absolute deviations from normal values. Nevertheless, in this case, measures would have different units (for instance Mbps and μs) and different meanings (which steams from the fact that, for instance, for some applications 5% of degradation on transit delay will be worst than 10% for other applications). Therefore, this type of measures cannot be compared and so they have limited interest. One alternative is to measure not the QoS deviations from normal values but their influence (or impact) on the entities related to QoS provision and consumption (the communication systems and the applications, respectively). Notice that with this approach the results can be made comparable and can consider different degrees of sensitivity to QoS variations. Our proposal to get meaningful QoS measures is based on this approach.

For short, our scheme to measure QoS turns possible to construct a unified view on the QoS actually provided by communication systems, despite the dissimilar nature of the objects to measure (as loss or delay) and the eventually distinct sensitivity to their variation. The information so obtained, we argue, will be very useful to anticipate problems in communication systems and to better tune their performance (or the performance of the mechanisms nuclear to their operation, as flow scheduling, policing, or admission control mechanisms).

3.1 Basic concepts and general assumptions

The first challenge to overcome in our proposal to measure QoS was to find a way to quantify the impact of QoS variations on the entities that deal with the affected flows. We use the concept of *measurement zones* (MZ) for this purpose. Before detailing it, it is important to note that the measures with higher granularity are the ones associated with a single flow QoS characteristic (for instance, the throughput or the transit delay of a given flow). From these ones it is possible to obtain other measures with broader granularity, as we will explain below. From now on, unless the contrary is explicitly pointed out, the presentation refers to measures of a single QoS characteristic.

According to the proposed metric, the value of a given QoS characteristic in a given communication system point may be within one out of four different zones: normal, degradation, superfluity, or intolerable zone (as depicted in Figure 1). These zones are named measurement zones, and are bounded by four values: the minimum (m) and maximum (M) values required for the QoS characteristic; and two threshold values (l_m or l_M) which define tolerable regions for QoS variation (the value O is the target QoS characteristic value).

Given the above, as depicted in Figure 2, the impact of QoS degradation on users is assumed to vary linearly[7] between:

- the null value – no impact, the QoS provided is in the vicinity of O;

- the value d – very low impact , the QoS provided is in the vicinity of m, and

- the value D – near intolerable impact, the QoS provided is in the vicinity of m-l_m.

Conversely, it is assumed that the impact on communication systems of QoS excess can vary through the values 0, $-d$ and $-D$.

Notice that the "sensitivity level" related to a given QoS characteristic variation is determined by l_m and l_M. For the same values of normal performance, the shorter is the degradation (or superfluity) QoS zone, the bigger will be sensitivity to QoS degradation (or excess). As measures reflect the impact of changing QoS on final entities that consume or provide it, they are naturally comparable (one of the major goals of this work). Given this, the challenge of our proposal became the definition of measurement zones for each flow's QoS characteristic and for each point of the communication system where we want to measure.

[7] It is possible to consider other types of variations – for instance, a logarithmic one, as proposed by Monteiro [Monteiro96, Monteiro95a], which is perhaps better suited to represent the sensitivity of human beings to QoS changes. Nevertheless, for now, and to avoid complexity, we will use the linear approach. Future work will address other types of sensitivity to changing QoS.

3.2 Establishment of measurement zones

The proposed metric assumes the possibility to establish a service contract per flow, containing, for each QoS characteristic, accepted end-to-end ranges, namely a minimum and maximum value. Our idea is to allow also the specification of the user sensitivity to the end-to-end QoS characteristic degradation, which is the degradation threshold l_m. To completely determine the end-to-end measurement zones we need also to define the superfluity threshold. In our proposal this threshold is calculated by the system, which is the penalised entity when QoS is provided in excess. As the superfluity threshold depends on the cost of the wasted resources, which, in turn, depends on the importance the resources have to users, its value is inferred from the degradation thresholds [Quadros98].

End-to-end measurement zones provide a way to measure variations of the QoS provided by the communication system, as the one presented in Figure 3. They provide a way to quantify the deviation of each QoS characteristic at its input and output, points P_t and P_r respectively. From those two measures it is possible to determine the performance of the communication system related to the considered QoS characteristic. For instance, if the measured QoS at the input is in the normal zone (which means it has a value between $-d$ and d) but in the degradation zone at the output (with a value between d and D), it is clear that the communication system is not conveniently supporting the considered QoS characteristic.

Nevertheless, the communication system shouldn't be seen as a whole. In many cases, it is more interesting to see it as a series of modules, with modules being end-systems or intermediate equipment, for example. Given this, in order to obtain the desired end-to-end QoS level, which we measure using end-to-end measurement zones, each module must provide a certain performance level. Again, to measure and control the performance given by each module we need to determine local measurement zones for each module, which must be calculated from the end-to-end measurement zones. The way to follow depends on the nature of the QoS characteristic, as can be seen in [Quadros98], which can be cumulative or non-cumulative. When cumulative characteristics are considered – as, for instance, transit delay – the end-to-end communication system performance corresponds to the addition of the performance given by each module. In this case, each module measurement zone will be a fraction of the end-to-end one. Conversely, when non-cumulative characteristics are considered – as, for instance, throughput – each module has equal performance responsibilities, so module measurements zones will be equal to the end-to-end one.

Summing up, when using the QoS metric we are proposing, after the establishment of a given flow a set of measurement zones will be installed in the modules along the flow path (using some protocol to establish flows, as RSVP). With them it is possible to measure QoS deviations from the expected values at the edges of the modules, and, from those measures, the module performances.

3.3 Types of Quantification

The metric we are presenting allows two types of quantification:

- the impact on the communication system or on its users, of QoS characteristics deviation from expected values (for different points of the communication system, for one or various QoS characteristics, and for one flow or a group of flows);
- the contribution of a communication module, or a set of contiguous modules, to the referred deviation or impact.

These types of quantification are materialised through two indexes: the *sensitivity index* and the *congestion index* respectively.

The sensitivity index has implicitly been referred before, it is calculated for a given communication system point and takes values from *-D* to *+D*, according to the deviation from the reference of the considered QoS characteristic (as depicted in figure 4). The congestion index embodies the evaluation of the communication module performance. It is calculated taking the sensitivity indexes at the input and output of the considered module.

Let q_j be a QoS characteristic of a flow F_i, $Is_{F_i,q_j}^{M,In}(t_k)$ and $Is_{F_i,q_j}^{M,Out}(t_k)$ the values of its sensitivity index at module M input and output, at instant t_k. If the QoS characteristic considered is a non-cumulative one, then the module M congestion index related to the same flow characteristic will be given by the following expression[8]:

$$Ic_{F_i,q_j}^{M}(t_k) = \begin{cases} Is_{F_i,q_j}^{M,Out}(t_k) & \text{if} \quad Is_{F_i,q_j}^{M,In}(t_k) \le 0 \;\vee\; \left[\left(Is_{F_i,q_j}^{M,In}(t_k) > 0\right) \wedge \left(Is_{F_i,q_j}^{M,Out}(t_k) > Is_{F_i,q_j}^{M,In}(t_k)\right)\right] \\ 0 & \text{if} \quad \left(Is_{F_i,q_j}^{M,In}(t_k) > 0\right) \wedge \left(Is_{F_i,q_j}^{M,Out}(t_k) = Is_{F_i,q_j}^{M,In}(t_k)\right) \end{cases} \qquad (1)$$

[8] Notice that the definition does not contemplate situations where the sensitivity index is worst at input than at output., which is only possible if the module is able to ameliorate the flow's QoS characteristic. This is, nevertheless, an open issue, which can be re-evaluated when considered convenient.

This formula states that the module *M* congestion index related to a flow non-cumulative QoS characteristic (such as throughput) equals the correspondent output sensitivity index, except in one case which corresponds to a null congestion index: the flow QoS characteristic has a worst than normal value at input and maintains this value at output.

If the considered QoS characteristic is cumulative then the module *M* congestion index is:

$$Ic^{M}_{F_i,q_j}(t_k) = (Is^{M,Out}_{F_i,q_j} - Is^{M,In}_{F_i,q_j}(t_k)) \times \rho_M \qquad (2)$$

with $\rho_M = {}^{O_M}\!/\!{}_{O_T}$, where O_M is the QoS characteristic target value for Module *M* and O_T the end-to-end QoS characteristic target value

Thus the module congestion index related to a QoS characteristic as, for instance, delay, corresponds to the difference between the output and input sensitivity indexes weighted by the fraction of the end-to-end QoS level that is due to the module in question.

As told before, measures related to a flow QoS characteristic have the highest possible granularity. But measures with larger scope are clearly needed. As we need to evaluate QoS given to flows taking into account more than one QoS characteristic (delay and losses, for example), it is interesting to have measures which consider several flow QoS characteristics as a whole. In turn, as we need to evaluate the global performance of the modules, which must contemplate all the flows they support, it is also important to have measures that consider not only one but several flows. In our proposal, from the measures related to single QoS characteristics it is possible to obtain measures with larger scope, corresponding to a group of QoS characteristics of a given flow, or to a group of flows.

At a certain instant of time, a module congestion index related to several QoS characteristics of a flow (or a flow's module congestion index) can be calculated averaging the module congestion indexes at that particular instant in time, considering each characteristic independently. Thus the flow's module congestion index related to a flow F_i at the instant of time t_k is:

$$Ic^{M}_{F_i}(t_k) = \frac{1}{n} \cdot \sum_{j=1}^{n} Ic^{M}_{F_i,q_j}(t_k) \qquad (3)$$

In turn, if we consider all the flows a given module supports at a certain instant of time we can determine the *global module congestion index* at that instant. Using the same reasoning as before, the global module *M* congestion index can be achieved averaging and weighting:

$$Ic_g^M(t_k) = \sum_{i=1}^{n} r_i \times Ic_{F_i}^M(t_k) \qquad (4)$$

This expression uses the concept of *flow relevance* (r). The concept results from the fact that different flows contribute dissimilarly to the module congestion, as they can use distinct module capacities. Assuming that the module resources used by a flow are directly related to the flow bandwidth, flow relevance can be defined as the quotient between the flow bandwidth and the module throughput[9].

Summing up, the metric defines a set of indexes that quantify the QoS deviation from desired values, providing a way to measure the performance of a communication system module when supporting a QoS characteristic of a flow, a group of QoS characteristics, or a group of flows. All those measures are comparable. This turns possible, for instance, to sort the deviations, or to find out which one is more acute. We argue that if absolute measures were taken we would not have this capacity to sort or to aggregate measures, and so the QoS quantification would be much less useful.

4. On-going Work

We have implemented the proposed metric in a FreeBSD system running as a router. The purpose of the prototype was to test the concepts, to analyse specific problems related to the implementation of our proposal and to measure the metric overhead. The reason why we have chosen FreeBSD steams from the possibility to have full access to its kernel source code and from its wide use for the development of QoS related software (for instance RSVP, which we are considering to use in our project).

The main implementation problems of our proposal were due to FreeBSD timing characteristics, which is not a real-time operating system. Because of this, we had some problems to control measurement events and to obtain timing information with adequate precision and without too much performance overhead. Despite these difficulties we have now a prototype able to perform per flow measurements related to transit delay and throughput (flows are identified based on the source and destination IP address and port). Thus, our FreeBSD implementation is simulating a communication system module where, somehow, measurement zones were installed and is able to calculate sensitivity and congestion indexes that provide information on the QoS that is actually provided to the flows supported by the module.

[9] Another approach is to let the system to define the *flow relevance* for each flow. If the used communication system allows priorities between flows, this could be an interesting alternative.

Presently, we are fine debugging and tuning the software. Nevertheless, we have already made basic performance tests to our implementation with good results. We measured the round trip time of probe packets sent from a machine *A* to a machine *B*, through the router under test, and back again to machine *A*. Naturally, we compared tests made when the router was running normal FreeBSD software and when it was running the version we modified to include our metric. We tested two different scenarios: (1) router without load; (2) router loaded with a flow of 512Kbps, corresponding to one packet of 100 bytes in every 1,56 ms. Our prototype was configured to measure QoS characteristics of probe packet flows, and the measurements were performed in time periods of 30ms, separated 150ms from each other[10].

The results obtained in our tests, made for probe packets of 64 bytes and 1000 bytes, are presented in table 4. They demonstrate that there is no meaningful difference between the performance of the normal FreeBSD kernel and the performance of the kernel modified to measure QoS according to our proposal. We concluded that our proposal can in fact be implemented without a prohibitive performance penalty.

As future work, we are planning to further develop our prototype in order to obtain a comprehensive system able to support end-to-end QoS flow needs (a system based on the UC-QoS model [Monteiro96, Quadros98]). For this we mainly need:

- to develop a way to install measurement zones in the different modules that make up the flow path (we are considering to use RSVP for this purpose);
- to develop an admission control algorithm based on our metric (namely, based on the congestion index of each module);
- to develop a flow scheduling mechanism based on our metric (we are planning to use flow congestion indexes to arbitrate the scheduling of flows).

We are also starting a project to develop a differentiated service router. In this case our plan is to use the TOS bits available in the IP header to classify packets in different traffic classes. Routers will have pre-installed measurement zones that will be used to measure delay and loss rates for the different traffic classes. The single difference between these zones will be their threshold values, as we are considering that the only difference between traffic classes is their sensitivity to transit delay and losses. With our metric we will continuously have information on the impact of eventual QoS degradation on the different classes.

[10] The prototype was designed to measure QoS characteristics during a configurable time period separated by time intervals which follow an exponential law. Nevertheless, for test purposes, the intervals between measurement periods had a fixed value.

So, we will be able to determine, at each instant, which class needs more attention from the scheduler mechanism. Summing up, using the proposed QoS metric it will be possible to dynamically adjust the priority given to flows, which may lead to an improvement of resource sharing and utilisation.

5. Conclusion

In this paper, we presented a comparative analysis of the main approaches that are currently being proposed as frameworks or models for the provision and guarantee of quality of service in communication services and systems. These are the ISO/IEC QoS framework, the IETF ISWG Integrated Services model, the QoS Architecture, the ATM traffic and congestion control model, and the Tenet model. The main goal of this analysis was to sum up the main characteristics of these models, and to identify the challenges - in the form of desirable capabilities - that should be addressed in order to construct QoS-capable systems.

It seems clear that one important component of any efficient communication system with QoS capabilities, deserving more research attention, is the monitoring function. In turn, an essential element of a monitoring function is a QoS metric that can gather significant and normalised information on the communication system behaviour in terms QoS provision. In this paper we have briefly proposed such a metric.

This QoS metric was implemented and a testbed was constructed, in order to prove the basic concepts feasibility. The first test results showed that the overhead of such a metric can be kept significantly low, which opened the door to a set of more elaborate experiments and projects in the area of quality of service guarantee.

References

[atmf-uni3.1] *ATM User Network Interface Specification*, version 3.1, ATM Forum, Sept. 94.

[Aurreco95] Cristina Aurrecoechea, Andrew Campbell and Linda Hauw, *A Survey of Quality of Service Architectures*, ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-95-18.ps.Z

[Banerjea94] Anindo Banerjea, Domenico Ferrari, Bruce A. Mah, Mark Moran, “The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences”, *IEEE/ACM Transactions on Networking*, vol. 4 (1), pp. 1-10, February 1994

[Braden94] Bob Braden, Dave Clark, Scott Shenker, *Integrated Services in the Internet Architecture: an Overview*, RFC 1633, IETF - Networking Working Group, May 94.

[Campbell94] Andrew Campbell, Geoff Coulson, David Hutchison, “Quality of Service Framework”, *Computer Communication Review*, Vol. 24 (2), pp. 6-27, Abril 94.

[DiffServ] http://diffserv.lcs.mit.edu/

[Ferrari94] Domenico Ferrari, Anindo Banerjea, Hui Zhang, “Network Support for multimedia, A discussion of the Tenet Approach”, *Computer Networks and ISDN Systems*, Vol. 26 (10), pp. 1267-1280, July 94.

[Gupta93] A. Gupta, W. Heffner, M. Moran, and C. Szyperski. "Network Support for Realtime Multi-Party Applications", In *Collected abstracts from the Fourth International Workshop on Network and Operating System Support for Digital Audio and Video*, Lancaster, England, November 1993.

[ISO/IEC9680] ISO/IEC JTC1/SC21 N9680, *Open Systems Interconnection, Data Management and Open Distributed Processing, Information Technology - Quality of Service - Framework*, Final CD, July 95.

[Kurose93] Jim Kurose, “Open Issues and Challenges in Providing Quality of Service Guarantees in High Speed Networks”, *Computer Communication Review*, Vol. 23 (1), pp. 6-15, January 93.

[Monteiro95a] Edmundo Monteiro, *Controlo de Congestão em Sistemas Intermediários de Camada de Rede*, Ph.D. Thesis, University of Coimbra, Portugal, 95. (portuguese)

[Monteiro95b] Edmundo Monteiro, Fernando Boavida, Gonçalo Quadros, and Vasco de Freitas, "Specification, Quantification and Provision of Quality of Service and Congestion Control for New Communication Services", In *Proceedings of the 16th AFCEA Europe Brussels Symposium & Exposition*, AFCEA/IEEE ComSoc, Brussels, Belgium, 18-20 October 95.

[Monteiro96] Edmundo Monteiro, Gonçalo Quadros, Fernando Boavida, "A Scheme for the Quantification of Congestion in Communication Services and Systems", In *Proceedings of the third International Workshop on Services in Distributed and Networked Envirnoments*, IEEE ComSoc, Macau, 3-4 June 96.

[Partridge93] Craig Partridge, “Protocols for high-speed networks: some questions and a few answers”, *Computer Networks and ISDN Systems*, Vol. 25, pp. 1019-1028, 1993.

[Quadros98] Goncalo Quadros, *A QoS Metric for Packet Switched Communications Systems*, Technical Report, University of Coimbra, January 98.

[Shenker97] Shenker, Wroclawski, *Network Element Service Specification Template*, RFC 2216, IETF-Integrated Services Working Group, September 97.

[Shenker97b] S. Shenker, C. Partridge R. Guerin, *Specification of Guaranteed Quality of Service*, RFC 2212, IETF-Integrated Services Working Group, September 97.

[Steinmetz97] R. Steinmetz, L. C. Wolf , "Quality of Service: Where are We", In *IWQOS’97 Proceedings*, New York, USA, May 1997.

[Wroclaw97] J. Wroclawski, *Specification of the Controlled-Load Network Element Service*, RFC 2211, IETF-Integrated Services Working Group, September 97.

	ATM 3.1	IS	QOS-A	TENET RMTP[11]	TENET CMTP[12]	ISO/IEC
Throughput	-Peak rate -Sustained rate -Burst length -CDVT	-Peak rate -Sustained rate -Burst length	-Peak rate -Sustained rate -Burst length	-Peak rate -Sustained rate -Burst length	-Peak rate -Sustained rate -Burst length	-Any type
Delay	-Maximum -Mean	-Throughput & Slack term -Class	-Maximum	- Maximum -Violation probability	-Maximum	-Any type
Jitter	-Max. (reference 1 or 2 points)	- Not specified	-Maximum	-Maximum	-Not specified (implicit. controlled)	-Any type
Loss	-Max. % cell not well received	-Implicit to the chosen service	-maximum loss per window time	-Probability due to congestion	-Granularity -Probability due to congestion or delay -Pattern bit to substitute error	-Any type
MTU specified?	-Not applicable	-Yes	-Yes	-Yes	-Yes	-Possible
mTU specified?	-Not applicable	-Yes	-No	-No	-Yes	-Possible

MTU - maximum transfer unit   mTU - minimum transfer unit

Table 1 - Specification of traffic and QoS characteristics

	ATM 3.1	IS	QOS-A	Tenet Scheme2	ISO/IEC
Possibility to negotiate the Service Contract	No	No[13]	Yes[14]	No	Yes
Possibility to specify not only objective values but also limits	No	No	No	Yes	Yes
Possibility to specify the Type of Service-TS	No	Yes	Yes	Yes	Yes
Possibility to specify the TS orthogonaly	N.A.	No	Yes	Yes	Yes
Possibility to explicitly specify statistical requirements	No	No	Yes	Yes	Yes
Possibility to specify actions to be triggered by QoS degradation	No	No	Yes	No	Yes
Possibility to establish the Service Contract in advance	No	No	Yes	Yes	Yes

Table 2 - Specification Characteristics

	ATM 3.1	IS	QOS-A	Tenet Scheme2	ISO/IEC
ossibility to have multicast connections with non uniform QoS	No	Yes	Yes	Yes	Yes
QoS established dynamically [D], or statically [S]	S	D	D	D	D
Possibility to associate priority to connections	No	No	No	No	Yes
Model deals with pricing?	f.f. study	f.f. study	f.f. study	f.f. study	f.f. study
Addresses security?	No	No	No	Yes	Yes
Integrated metric to evaluate QoS	No	No	No	No	No

Table 3 - Communication systems characteristics related to QoS

[11] Real-Time Message Transport Protocol
[12] Continuos Media Transport Protocol
[13] Depending on the establishing protocol and on the used services, it should be possible to negotiate the installation of the service contract, in the case of the IS model

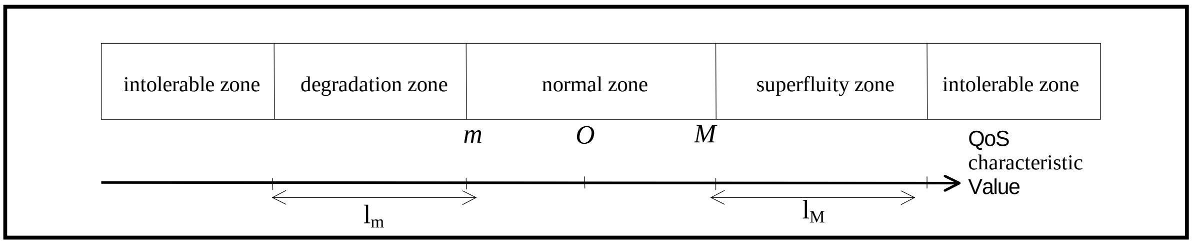

Figure 1 - Measurement zones

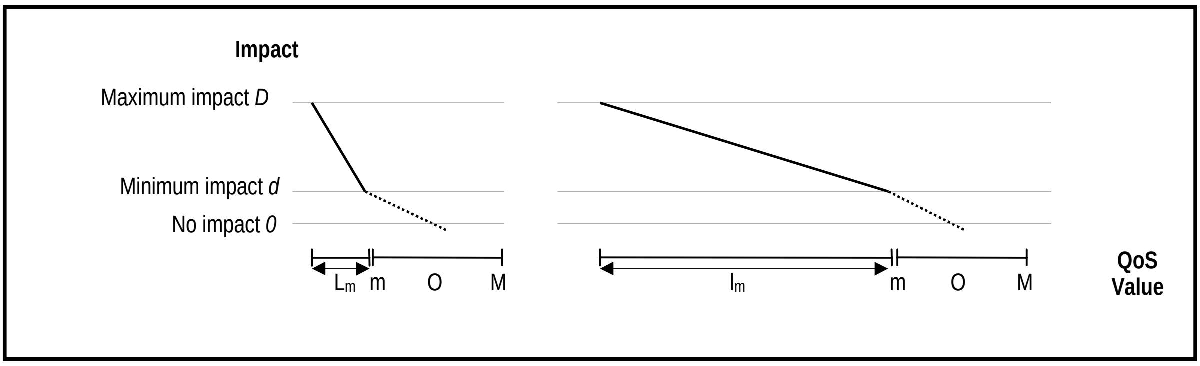

Figure 2 - The degradation zone length, l_m, allows the specification of a higher (left) or lower (right) sensitivity to QoS degradation

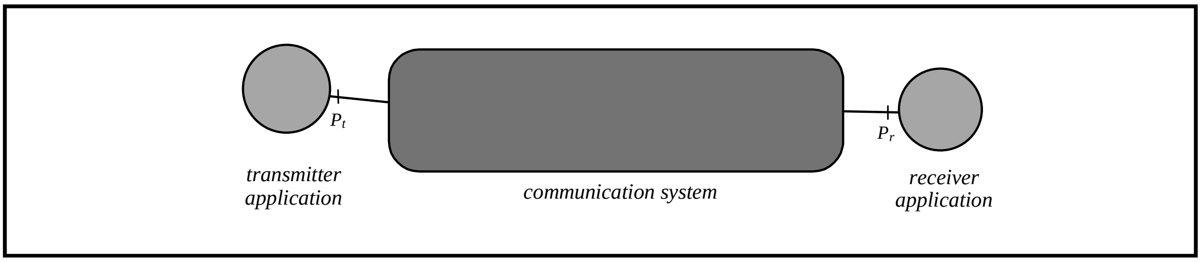

Figure 3 - A simple communication system model

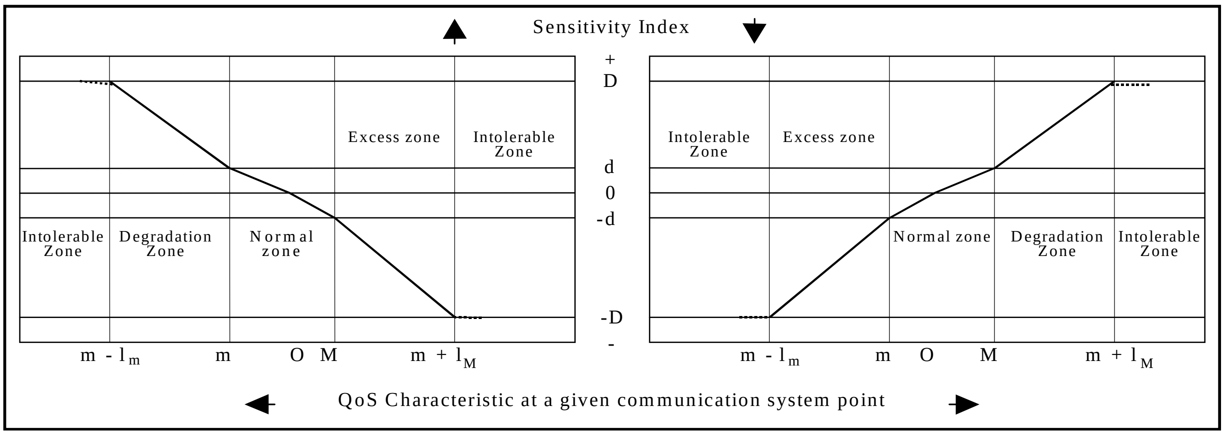

Figure 2 - Sensitivity index evolution in different functioning zones
left part – degradation for values below *m* (as is the case of throughput)
right part – degradation for values above *M* (as is the case of transit delay

[14] QoS-A enables alternative QoS specifications to be used in case of violation of any of the original values. This can be seen as a way to specify minimum limits for QoS characteristics.

Load Kbps	Pck. Lenght	Modified Kernel				Normal kernel			
		Min	Average	Max	Std. deviation	Min	Average	Max	Std. deviation
No	64B	0,674	0,725	3,4325	0,265	0,668	0,681	1,029	0,029
No	1000B	4,235	4,262	6,863	0,1335	4,223	4,224	4,674	0,039
512	64B	0,643	3,486	14,788	2,925	0,635	3,082	13,367	2,497
512	1000B	4,209	6,668	19,387	2,553	4,200	6,402	15,393	2,252

Table 4 – Results of the performance tests made to our prototype. Each line corresponds to the average of several experineces; each experience involves 500 probe packets.