INTEGRATED QUALITY OF SERVICE FOR MULTIMEDIA COMMUNICATIONS

Andrew Campbell, Geoff Coulson, Francisco García,
David Hutchison and Helmut Leopold[†]

Department of Computing,
Lancaster University,
Lancaster LA1 4YR, U.K.
E.mail: mpg@comp.lancs.ac.uk

[†]Alcatel ELIN Research,
A-1210 Vienna,
Ruthnergasse 1-7, Austria.
E.mail: Helmut.Leopold@rcvie.co.at

ABSTRACT *The integration of distributed multimedia systems support into a communications architecture encompassing the new multiservice networks poses significant challenges. A key observation about the new environment is that Quality of Service (QOS) provides a unifying theme around which most of the new communications requirements can be grouped. For applications relying on the transfer of multimedia information, and in particular continuous media, it is essential that QOS is guaranteed system-wide, including the distributed system platform, the transport protocol and the multiservice network. Enhanced protocol support such as end-to-end QOS negotiation, renegotiation, indication of QOS degradations and co-ordination over multiple related connections are 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. This paper describes the first stage in the development of an integrated Quality of Service Architecture (QOS-A) which offers a framework to specify and implement the required performance properties of new multimedia applications over multiservice ATM-based networks.*

1. Introduction

The evolution of distributed computing is being simultaneously influenced by the emergence of high-speed multiservice networks and the requirements of new distributed multimedia applications [1]. Due to the requirements of distributed multimedia applications, future communications infrastructures should offer comprehensive QOS configurability for a wide variety of media types and specific user requirements. An important first step in meeting such a requirement is the specification of a Quality of Service Architecture (QOS-A) which offers an integrated framework for QOS specification and resource control over all architectural layers from distributed application platforms to the network layer.

The approach of this paper is to present a set of key QOS requirements and map these requirements onto a provisional QOS-A which has emerged from an experimental system designed and implemented at Lancaster. Because of the likely complexity of a fully general QOS-A, we limit the scope of our discussion in this paper to aspects of a QOS-A for the support of continuous media communications. We also concentrate on ATM at the network layer rather than consider the full range of multiservice networks. However, a generalised QOS-A should eventually be extensible to incorporate other areas of QOS provision such as real-time control systems, file transfer and real-time transaction processing.

The paper is structured as follows. Section two motivates the requirement for a QOS-A in the light of the emerging requirements of distributed multimedia applications. Section three reviews current notions of QOS in OSI and the current ATM proposals from CCITT. Section four then presents research at Lancaster which has defined a baseline QOS architecture. In addition, this section also looks at functions and mechanisms for QOS support and attempts to place them within the evolving QOS-A. Finally, section five briefly examines related work in the field and section six presents our conclusions.

2. The Need for a QOS-A

2.1 QOS Requirements for Continuous Media

With the emergence of multimedia information exchange, increased requirements are placed upon communications support. Multimedia is particularly characterised by *continuous media* such as voice, video, high quality audio, and graphical animation, which place much greater demands on communications than still media, such as, text, images and graphics. Different types of continuous media require different levels of latency, bandwidth and jitter, and they also require guarantees that levels of service can be maintained. For example, video connections require high throughput guarantees but telephone audio requires only modest bandwidth. Error control should also to configurable: e.g. uncoded video is highly tolerant of communications errors whereas compressed voice can tolerate almost no errors and file transfers should be 100% error free. Delay jitter (i.e. variance in delay) is an additional factor which must be taken into account for continuous media transfers and must be kept within particularly rigorous bounds to preserve the intelligibility of audio and voice information.

In distributed multimedia applications the concept of QOS becomes applicable on a full *end-to-end* basis. In addition to the communications sub-system, this has implications for operating system scheduling for threads which are producing/ consuming information for quality controlled connections. End-to-end QOS also involves *distributed application platforms* which are layered on top of the operating system to provide distribution transparencies and object based computational models for the benefit of programmers of distributed multimedia applications.

Finally, the concept of QOS is also applicable to areas other than the traditional arena of point-to-point connections. For example, the prevalence of multicast and group communications in distributed multimedia systems leads to considerations such as the ordering semantics of group message delivery which can be treated as a QOS issue. Also, the requirements of multimedia synchronisation such as ‘lip-sync’ impose QOS constraints over multiple transport level connections. We refer to this latter requirement as *orchestration* (see later); QOS properties in this context are concerned with the ‘tightness’ of orchestration required and the strategies to adopt when QOS provision degrades.

2.2 QOS-A Requirements

Because of the increased range and complexity of QOS provision required by the emerging distributed applications, it becomes essential that the necessary extensions to QOS provision are not done on a piecemeal basis. Instead, we advocate the notion of a comprehensive QOS-A, whereby application requirements can be mapped through all the levels of the system. Thus, communications abstractions at the application platform level should provide QOS abstractions which can be mapped down through all intermediate layers to the multiservice network access point in a coherent and integrated way. This mapping should be clean, simple and efficient, protecting application programmers from communication details; that is, they should be stating what they require rather than how it is to be achieved.

In addition to mapping functions, the QOS-A should provide a framework for the support of QOS throughout the layers. In particular, there is a need for enhanced protocol support which is lacking in current protocol specifications. The framework will include *management functions* and selectable QOS support *mechanisms*. Examples of management functions are *i)* end to end QOS negotiation including admission control for new connections, *ii)* policing to ensure that users are not violating negotiated QOS parameters, and *iii)* monitoring to ensure that negotiated QOS levels are being maintained by the service provider;

QOS support mechanisms take the form of a pool of available procedures which can be configured and inserted into a protocol stack. Depending on the QOS contract established at negotiation time, the QOS-A would be responsible for building a suitable stack profile by selecting from the available set of mechanisms. Examples of such mechanisms are error control modules, parameterisable scheduling modules and jitter smoothing modules. Once again, many of these mechanisms will be applicable at multiple system layers. For example, scheduling appears in the context of end user threads and also in network switches.

3. Current Notions of QOS

Traditionally, the term ‘quality of service’ in the communications context referred to certain characteristics of network services as observed by transport users. These characteristics were not controllable by users, and described only those aspects of services attributable to the network provider. QOS parameters in current communications infrastructures, like OSI and CCITT do permit the specification of some user requirements but these are almost never supported by the underlying network. For example, the current OSI standards treat QOS in a layer specific way, and QOS definition has been looked at by separate committees (i.e. the presentation, session, transport, network and data link committees) working in isolation. Thus the relationship between QOS layers is not clearly defined and there is no consistent, integrated notion of QOS which relates user requirements to the network provider services.

The following sub-sections review the degree of QOS provision in sample architectures at the session, transport and network layers. The session and transport layer specifications are taken from the ISO’s Reference Model for Open Systems Interconnection (OSI-RM) and the network specifications are taken from the CCITT’s series I recommendations for ATM cell switching [2].

3.1 OSI Perspective

In the OSI-RM, QOS parameters associated with the application layer’s P-CONNECT primitive are generally mapped directly down to the associated QOS parameters at the session layer. Thus the QOS parameters associated with a P-CONNECT service element exist solely to give the application process access to the corresponding parameter of the session service element, S-CONNECT. The functionality of the session layer QOS parameters is then mainly concerned with monitoring and maintaining session services to a level agreed by the negotiation between peers as part of connection establishment.

There are, however, aspects of the S-CONNECT QOS parameter which relate directly to the reliable data transfer environment required to service the session. These aspects are included in the QOS parameter of the transport layer T-CONNECT service element. The session layer QOS parameter is in fact a list of parameters, each of which relates to a particular QOS performance parameter. There are parameters covering each of the phases of the session; i.e. connection establishment, data transfer and connection release. The parameters are also classified into two major groups: *performance oriented* and *non-performance-oriented*. The non-performance-oriented parameters do not directly effect the performance of the communications but are concerned with protection, priority and cost aspects. The performance parameters include: throughput, transit delay, residual error rate, establishment delay, establishment failure probability, transfer failure probability, resilience, release delay and release failure probability.

3.2 CCITT I-Series Perspective

The CCITT, in their series-I recommendations, have recognised the need for QOS configurability in the emerging ATM standards for B-ISDN and a fairly comprehensive set of parameters has been defined. QOS of bearer services in ATM networks is applicable at three control levels:-

- *call control level*: this is concerned with the establishment and release of the call. A call is rejected by the call acceptance control algorithm if the requested bandwidth is not available at the time of call set-up request;
- *connection level*: this is concerned with allocation of resources for the data transfer phase. A call is rejected if there is no available path (sequence of links) to its destination. At the connection level resources have to be allocated at each intermediate hop between the source and the destination;
- *cell control level*: this is concerned with the data transfer phase itself. Once a connection has been accepted, the cell stream must be policed to ensure that the user does not exceed the values contracted in the call-setup;

A user wishing to establish a connection signals his QOS requirements to the ATM network. The signalling message includes a declaration of the QOS characteristics of the user data which have somehow been mapped down from the higher layers, and enables the connection acceptance control function to allocate the required QOS resources if the connection is accepted. The connection is then assigned a source policing function which monitors the cell stream and causes cells exceeding the declared traffic rate to be either discarded immediately or marked to be discard later if necessary.

At the call control level, the available parameters are similar to those defined in OSI: i.e. establishment delay, establishment failure probability, release delay etc.. At the connection level the parameters include: peak arrival rate of cells, peak duration, average cell arrival rate, burstiness, cell loss ratio (CLR), cell insertion ratio (CIR) and bit error rate (BER).

The cell level employs the traffic characterisation supplied by the connection level and uses it to ensure that the application does not exceed the peak and average traffic levels agreed at connection time. As an example of traffic characterisation, the QOS parameters of variable bit rate encoded video could be: peak rate = 50Mbps, average cell arrival rate = 25Mpbs, burstiness = 2 and the peak duration = 10ms.

3.3 Evaluation

The clearest point to emerge is that QOS is currently looked on largely as a service provider issue whereas the requirements identified in section 2 imply that QOS should also be a user level issue. It is also clear that the OSI standards are currently incomplete and inconsistent for specifying QOS properties. In particular, the protocol specifications and service definitions do not include any notion of QOS management and the semantics of responsibilities and guarantees are not clear. Furthermore, those functions which are defined are almost never supported by protocols and networks.

Another important point is that the OSI upper layers have no notion of QOS: QOS parameters are simply mapped through to the transport layer. This is also true in alternative upper layer architectures such as the object-based ODP architecture [3]. If users want to specify QOS they are forced to drop below the level of abstraction provided by these architectures and interact with layers that are supposed to be hidden. Furthermore, there are very limited facilities for QOS negotiation at the user level. A user must simply specify the parameter values required and let the lower layers either accept or reject the proposal.

The CCITT's ATM recommendations are more comprehensive in scope with a fairly detailed traffic characterisation model. Here it is the mapping between the higher layers and the ATM adaptation layer, and also the mechanisms required to support particular QOS specifications which are lacking in substance. These are precisely the concern of a generalised QOS-A. An important step in our work will be resolving the present inconsistencies in the relationship traffic characterisation parameters of ATM and the OSI-RM. Other requirements are the development of protocol support for QOS in terms of the various QOS management functions and support mechanisms, examples of which were given in section two above.

Finally, a major limitation of all current notions of QOS is that the value of a QOS parameter, whether negotiated or not, remains the same through the lifetime of a connection: i.e. once negotiated a QOS parameter is never re-negotiated. Another implication of this is that the service-provider is committed to provide the QOS over the lifetime of the connection. There is, however, no guarantee that the service-provider will be able to maintain the originally specified values: in fact maintaining end-to-end service levels in the face of variable load is an unsolved problem involving resource scheduling at multiple levels. Even when the QOS of a connection does deteriorate the service provider is under no obligation to signal such a change in QOS to the users of the connection. The provider may, however, disconnect the connection unilaterally.

The essential characteristics of the current state of QOS provision may therefore be summarised as follows:-

- *lack of overall framework*: the framework for QOS must extend from the distributed application platform through the transport subsystem and the network. It must also encompass QOS considerations in areas such as orchestration and groups;
- *inconsistency*: the framework must build on and reconcile the existing notions of QOS particularly in the OSI-RM and the ATM series I recommendations;
- *incompleteness*: the framework should include extensions to current QOS provision as detailed in the following section;
- *lack of mechanisms to support QOS guarantees*: research is needed in basic mechanisms such as scheduling so that contracted QOS levels can, in fact, be maintained.

4. An Extended View of QOS

4.1 Baseline Architecture

4.1.1 Architectural Layers

The baseline for the development of the QOS-A is the layered architecture depicted at the left hand side of figure 1. This has been derived from our experimentation to date with distributed multimedia applications. A detailed description of our current infrastructure and its implementation is given in [4]. The remainder of figure 1 illustrates the aspects of the QOS-A to be described in this section.

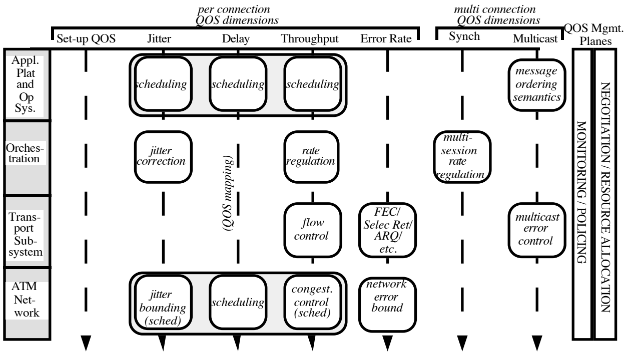

Figure 1: QOS-A

The upper layer in the layered architecture consists of distributed applications platform which is provided by an ODP compatible distributed systems platform augmented with services to provide multimedia communications, QOS configuration and synchronisation [4].

Below the platform level is a layer of services used to add value to the functionality provided by the lower transport layer. Specifically, these services control jitter and rate regulation for continuous media streams. They also provide these services, together with cross-stream synchronisation, across multiple application related connections. Because these services are concerned with co-ordinating multiple sources and sinks we refer to them as *orchestration* services; a full description of these services can be found in [5].

Below the orchestration services is a transport service and protocol which is specifically designed for continuous media communications. It is highly configurable in terms of QOS and offers full end-to-end QOS negotiation and re-negotiation. Full details of the transport services are available in [6].

The communications infrastructure is provided by a multiservice network. Currently we are using a real-time FDDI emulation, but are in the process of upgrading the communications to use an ATM switch. To achieve this aim, we require new hardware interfaces to our current multimedia workstations, and also an implementation of the ATM adaptation layer software.

4.1.2 QOS Dimensions

In figure 1 we have attempted to extract a canonical and orthogonal set of *dimensions* within which traffic can be characterised in our chosen domain of continuous media communications. The chosen set of dimensions are: *set-up*

QOS (i.e. the OSI establishment and release parameters), *jitter* (i.e. variation in delay), *delay*, *throughput* and *error characteristics*. In addition to these fundamental dimensions, two additional dimensions are included: *synchronisation* between media streams, and aspects of *multicast* quality of service. The essence of these latter two dimensions is that they are applicable over multiple connections whereas the others apply to single connections. In fact, the multi-connection dimensions also subsume the fundamental dimensions but additional quality of service characteristics arise as emergent properties.

Later sections describe how traffic may be specifically characterised at the various layers, and how levels of service along the canonical QOS dimensions are maintained through profile selection at the different layers.

4.2 QOS Management Functions

4.2.1 QOS Negotiation

The most fundamental aspect of the QOS-A is the interface at which desired levels of QOS can be requested, negotiated and contracted. In a layered architecture such as figure 1, there are multiple instances of this interface; each instance has a user above the interface and a provider below. The function of the QOS-A here is to permit end-to-end QOS negotiation from the top user level down to the network layer and up again at the remote site. A successful negotiation at each interface level results in a contract with two major clauses:-

- an agreed *level of service* which the provider level must undertake to maintain, and
- an agreed *level of traffic* which the user level must undertake not to exceed.

Both the level of service and the level of traffic will be expressed in terms of a common, layer specific, *traffic characterisation* language based on the fundamental QOS dimensions. Examples of layer specific traffic characterisations appear in section 4.3.

A further aspect to the contract is the *degree of commitment* in the above clauses: e.g. is the provider committed to maintaining the level of service in all conditions or are there circumstances in which the level of service may be relaxed? A related question is what sanctions will be imposed by the provider if the contracted traffic level commitment is exceeded by the user? Both of these points can also be related to the *cost* of the service; presumably a higher commitment by the provider for the same nominal service will cost more, as will the option of a lower commitment from the user.

To express degrees of commitment either a relative measure such as priority levels can be used, or absolute measure such as a percentage. Absolute measures can also be expressed as a step function with values such as {*deterministic, probabilistic, best-effort*}. An absolute scale is also appropriate for cost measures. Even if only a step function was ultimately available at the bottom level, percentages measures of commitment at higher levels could be used to express trade-offs. For example, in a videophone connection consisting of separate video and audio channels, a slightly lower commitment probability for the video channel would be appropriate Even if the two probabilities chosen mapped to the same step function value at the lower layer (e.g. probabilistic), the higher commitment of the audio channel would ensure that if one channel had to be taken down due to lack of resources, the more important audio channel would be preserved.

As pointed out in section 2 the same QOS interface is not necessarily appropriate for all layers in a layered architecture. However, in abstract terms, the interface at each layer will consist of some subset of the following general components: i) a notation for quality characterisation along the QOS dimensions, ii) a notation for commitment specification, iii) a notation for cost specification, and iv) a protocol for QOS negotiation.

There are other possibilities in the negotiation process which may be of use at different layers. For example, rather then simply proposing a level of service, upper and lower bounds on acceptability could be proposed by the user and the provider could return a contract as close to the upper bounds as possible. Also, renegotiation of the QOS on a live connection could be permitted. Finally, the user could specify alternative *degradation paths* to be taken when commitments are not met by the provider. Possible alternatives are: ignore the situation, simply inform the user, or inform the user and reconfigure according to a user specified degradation path. As an example of the latter a hifi audio channel could be degraded to a 64Kbps voice audio channel. At the same time, the system would inform the user who would adjust the source and sink codecs as appropriate.

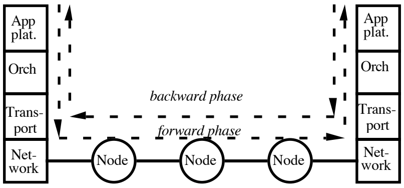

Figure 2: End-to-end QOS Negotiation

The mechanism of negotiation is illustrated in figure 2. In broad terms, negotiation is a two phase process. On the forward phase, from the source towards the sink, each intermediate resource holding node or end-system layer attempts to allocate resources to the request. Each layer in the system contains an *admission control* module which determines whether or not the request should be considered. For example, the operating system will decide if it is able to create a new real-time thread and the network layer will determine whether or not it can allocate a new connection. If the request is accepted by admission control, as many resources as are available are dedicated to the request. Eventually, at the sink end, the allocated QOS is compared against the requested requirement along each QOS dimension and the amount of over-commitment, if any, is calculated. On the second phase, from sink to source, any over-commitment is divided among the intermediate nodes and the over committed resources are released. An example of a negotiation mechanism from the literature is reported in [7].

4.2.2 Monitoring and Policing

QOS must be monitored at all layers of the QOS-A to ensure that the negotiated levels of QOS are being maintained. Each layer will collect statistical information associated with the on-going connection performance and make an assessment of the QOS measured against QOS requested. Monitors may then either attempt to take action to restore QOS levels themselves or they may choose simply to inform the upper layer that there is a problem.

Policing, on the other hand, is primarily seen as a network (and perhaps transport) function and may not be carried out at all layers. In particular, the application platform and orchestration layers are simply not able to violate QOS agreements by the nature of QOS support at those layers. Examples of monitoring and policing at the various levels in figure 1 are as follows:-

Application platform An important consideration at the platform level is the monitoring of end-to-end communications and will thus take into account possible QOS degradations arising from the end-system in addition to those arising from the communications subsystem. Application platform level monitoring will be mainly concerned with monitoring the performance of the operating system thread scheduling mechanism to ensure that data which arrives correctly is also delivered correctly to its final destination.

Orchestration A fundamental part of orchestration is the ability to monitor the on-going temporal relationship between of connections, and to regulate the connections to perform fine grained corrections if synchronisation is being lost. It is almost inevitable that related connections will eventually drift out of synchronisation due to such factors as discrepancies between remote clock rates and network congestion caused by temporary blocking at intermediate switches. The degree of multimedia synchronisation accuracy required by the user is viewed as a monitoring issue in the QOS-A. The monitor may choose to attempt to correct the drift (perhaps by renegotiating transport QOS), or if the drift is too great it may simply inform the upper level.

Transport The transport level monitoring entity captures statistical information related to the connection performance between two TSAPs. In the case of guaranteed QOS, the QOS state for each parameter is periodically measured to determine the *(i)* typical response time; *(ii)* average throughput over an interval *(iii)* amount of time buffering/blocking in the network per packet; *(iv)* worst case responses time; and *(v)* best case response time. If any of the negotiated QOS parameters degrade below the specified minimum tolerance, then a *QOS.Indication* [5] is raised by the monitoring mechanism detailing which QOS parameter has degraded and its current measured value, the indication also includes a statistical profile of all monitored parameters. On receiving a QOS.Indication, the application is free to make a judgement based on the current connections performance; for example the application may accept a poorer QOS in the light of network congestion, on the other hand it may initiate a full end to end renegotiation and also upgrade the level of commitment of the connections QOS.

Network Once a connection has been successfully negotiated between two end points, the source must be policed at the edge of the network to ensure that it does not exceed the traffic profile declared at ATM call-setup time. This is particularly important if a statistical multiplexing approach is assumed. Should the user exceed the agreed throughput levels, the policing function will intervene and discard cells. In addition a network level QOS.Indication will be generated indicating a *bandwidth violation* has been detected on the connection. On reception of the indication, the application can either regulate its traffic or renegotiate a higher throughput to meet its end to end throughput requirement. In a logical sense, monitoring is also required at the network level to ensure that the user is receiving the negotiated level of service. However, we envisage that this will mainly be the responsibility of the transport. Broadly speaking, the transport will monitor and the network will police.

4.3 Layer Specific QOS Considerations

4.3.1 Application Platform Layer

4.3.1.1 Functions

In our experimental configuration, this layer includes both the operating system and a distributed application platform based on the ANSA architecture [8]. In ANSA programmers view an object-based computational model in which all interaction is specified in terms of RPC invocations of named operations in location independent abstract data type interfaces. Thus it appears that QOS requirements should be attached to interface descriptions so that constraints on the QOS of invocations on that interface can be specified. The language used to specify such constraints should be network independent and its expression should be in terms of user level concepts such as frames rather than packets or bits.

Another consideration is that application programmers may not wish to be concerned with the precise details of a possibly complex set of QOS parameters. There should be a shorthand way of specifying commonly used ‘channel types’ by name such as HifiAud or StdVid. These standard channel types could also be parameterised with arbitrary semantics: e.g. Monochrome-Video(MPEG, 512, 256) would mean that the data stream will be compressed according to the MPEG standard and the throughput can be deduced from the required window size of 512 x 256 pixels. Such shorthand specifications could also implicitly include default QOS parameter ranges, probabilities of commitment and degradation paths as described above. As long as a common underlying QOS representation is used, new QOS channel types can be provided simply by adding new *QOS-mapper* services to the running system. These are run time services which resolve channel type names and parameters into the representation used at the next level down. A sample set of commonly used QOS channel types [1] are illustrated in table 1.

Channel Type	*Bandwidth*	*Jitter*	*Delay*	*Traffic type*	*Error*
StdVid	25 Mbps	10 ms	250 ms	prob	10^{-3}
SlowScanVid	1 Mbps	10 ms	250 ms	prob	10^{-2}
MPEGVid	10 Mbps	1 ms	250 ms	determ	10^{-9}
VoiceAud	64 Kbps	10 ms	250 ms	prob	10^{-1}
HiFiAud	2 Mbps	5 ms	500 ms	determ	10^{-5}

Table 1: Standard Channel Types

4.3.1.2 Mechanisms

The QOS support *mechanisms* at this level are almost entirely concerned with scheduling at the operating system level. As can be seen from figure 1, operating system scheduling effects the QOS dimensions of jitter, delay and throughput. However, this layer is not involved in QOS support mechanisms for error handling, synchronisation or multicast.

The way in which scheduling is parameterised to support the jitter delay and throughput dimensions is dependent on the particular scheduling policy used. In our system we are using a modification of the deadline/workahead scheduling implementation developed at the University of California at Berkeley and described in [9]. This scheme applies earliest deadline first scheduling to packets generated by devices or arriving from network connections according to a deadline timestamp contained in each packet. If the scheduler has sufficient processing resources, it is able to schedule the threads expecting these packets such that the packet deadlines are not violated. The scheduler is also careful not to schedule a thread too early and thus is able to control jitter.

The platform level is also involved in the QOS of multicast connections. At this level, multicast appears in the computational model as the invocation of groups of interfaces. A range of qualities of service are possible here most of which pertain to message ordering guarantees at the multiple interfaces. For example, *virtual synchrony* is a particularly strong QOS where it is guaranteed that if there are multiple clients of an interface group, all messages from the different clients are delivered to all the service interfaces in the same order. This also illustrates the general principle that a QOS specification determines the construction of a tailored 'profile' which includes modules such as virtual synchrony if required. The same principle is applicable to all the layers.

4.3.1.3 Example

Finally, as an example of platform level QOS we present the specification of a video channel in terms of an ANSA interface specification. This example will be taken up again as we examine the lower system layers in subsequent sections.

```
TYPE VideoSource = INTERFACE
BEGIN
GetVideo : OPERATION = [ ]
         RETURNS [ VideoFrame ]
         WITH QOS "StdVid";
END.
```

4.3.2 Orchestration Layer

4.3.2.1 Functions

Orchestration services provide value added communications services approximately at the session layer in OSI terms. However, the functionality of these services is not concerned with the traditional OSI telematics session functions. Instead, it offers services which are of use to continuous media applications: primarily rate control and jitter correction. The orchestration services also operate over multiple connections and provide synchronisation between continuous media streams flowing in separate connections. An example of such synchronisation is lip-sync between audio and video channels. To support such synchronisation relationships between channels, the orchestration services apply a compatible rate control to the flow of data in each connection and adaptively adjust these rates when brief disruptions occur. For example, if a number of packets are lost from an audio play-out, a corresponding number of packets may be dropped from an associated video channel.

The degree of multimedia synchronisation accuracy required by the user is viewed as a QOS issue in the architecture. This is expressed by the user at a platform level interface using the concept of per stream intervals which are interpreted as synchronisation points. A global clock service ensures that intervals can be interpreted similarly throughout the network. Other QOS concerns are the degradation paths to be taken when synchronisation is lost. These may include requests to *renegotiate* the QOS of transport connections or the adoption of longer intervals. A full description of the motivation for the orchestration services is beyond the scope of this paper. The same is true of the mechanisms used in orchestration. Full details may be found in [5].

4.3.2.2 Example

To return to the example above, our video QOS may be expressed in the following terms at the orchestration layer.

```
Delivery rate: 25 frames/sec
Permissible jitter: 10 ms
Sync interval: 1 second
```

4.3.3 Transport Layer

4.3.3.1 Functions

Traditional transports such as OSI Transport Class 4, have mainly serviced the communication needs of file transfer and low-bandwidth interactive applications. It is generally accepted that these protocols are not suitable for multimedia and real-time communications, and that dedicated transport services and supporting protocols are needed to exploit the performance of the new high speed network environment such as the ATM based multiservice networks [6]. Consequently, the following enhanced transfer services have been identified: *(i)* continuous media service; *(ii)* transaction oriented service; and *(iii)* bulk data transfer service. These transfer services should operate over connection-oriented and connectionless services and should also permit multicast topologies.

Continuous media transfer requires new transport mechanisms that support the following functionality:-

- QOS negotiation is required across all the QOS dimensions in figure 1 except for the cross stream synchronisation dimension;
- indications of QOS degradation to the upper layer are required. This is an aspect of monitoring discussed in section 4.2.2.
- re-negotiation of QOS on live connections is required. For example, users may wish to re-use the same connection for different media to minimise connection establishment delays;
- a flexible approach to error recovery. This is needed because error recovery needs vary considerably with the encoding used for continuous media data. For example, uncompressed video is highly tolerant of errors whereas MPEG compressed video is highly intolerant. A further consideration is that error recovery must not involve retransmission as packets will then arrive to late to be useful. For this reason forward error correction (FEC) is a promising technique here.
- multicast;
- simplex connections are useful as resources do not have to be allocated for full duplex operation when, say, a one way video transmission is required.

Lancaster's experimental *transport protocol* is a rate-based transport mechanism which supports the communications needs of continuous media. Currently it runs over the FDDI emulation mentioned above but is being extended to interface to the ATM adaptation layer (AAL). The transport layer defines the QOS performance in terms of parameters comprising burst size, burst duration, delay, jitter, priority, packet error rates and error profile. See [6] for further details.

4.3.3.2 Mechanisms

As can be seen from figure 1, the QOS dimensions applicable at the transport layer are throughput, error correction and multicast issues. Jitter and delay are not treated within the transport layer and these parameters are simply mapped through to the network layer. Throughput is configurable in the transport layer by manipulation of buffer resources and flow control mechanisms. To avoid the overhead of window based flow control, our experimental protocol uses rate based flow control which is well suited to continuous media data. Throughput is affected by the burst size and burst duration in the rate control scheme.

For errors, a variety of error correction profiles such as FEC, selective retransmission or go-back-N can be chosen depending on the QOS required. In multicast mode the error control dimension is expanded to include considerations of differential packet losses visible at each destination. 'Leaky bucket' error correction algorithms can be brought in to the profile to provide reliable multicast delivery if such a QOS is required.

4.3.3.3 Example

Returning again to our example, the QOS specification at the transport layer may be as follows.

```
Burst size: 100 Kbps
Burst rate: 100 per sec
Delay: 1 sec
Jitter: 20 ms
Priority: 10
Error profile: FEC
Error rate: 2%
```

Note that the jitter specified at this level is coarser than the top level requirement as the orchestration layer is taking some responsibility for jitter control.

4.3.4 Network Layer

4.3.4.1 Functions

The key requirement at the ATM layer is to achieve a high degree of resource utilisation by statistically multiplexing traffic while simultaneously meeting the users source traffic QOS requirements. Because of this requirement QOS at the ATM level is more concerned with traffic characterisation for efficient resource management than for user convenience. Note that this difference in perspective is also apparent in the emphasis on source traffic policing rather than service level monitoring. Because the ATM QOS parameters listed in section 3.2 have been formulated with this aim, and also allow QOS expression over the full set of QOS dimensions described in section 4.1.2, they are probably sufficient regardless of the traffic types involved.

One traffic characterisation issue still to be addressed at the ATM layer is the issue of level of service commitment. For ease of resource management, it is convenient to partition commitment into a small number of fixed levels rather than a continuous scale [10]. A possible partition is as follows:-

- *deterministic service*: the highest priority service the ATM network provider can support. Typically, this is a guaranteed QOS for hard real time performance applications;
- *probabilistic service*: this may suffer from QOS degradation from time to time because of the statistical nature of the network service. This service is particularly suitable for many continuous media applications that can compensate for loss of QOS because most media has built in redundancy. QOS monitoring, as described in section 4.2.2, is an important aspect of this service;
- *best effort service*: the lowest priority service, is synonymous with datagram services. No network resources are allocated or monitored because the network provider is not obliged to guarantee any level of service. This commitment level only receives whatever network resources are available after the other levels have been serviced.

4.3.4.2 Mechanisms

As is clear from figure 1, a network level traffic scheduling mechanism is an central component of the QOS-A as this directly supports bounded delay, jitter and guaranteed throughput. Scheduling is important because the dominant factor in delays and jitter is the network queuing delay introduced by intermediate switches rather than the propagation delay between switches. The network must provide strong performance guarantees for deterministic and probabilistic traffic (even during peak-load and over-load conditions) by correctly scheduling cell departure times at the end-systems and at each switch so that delays can be kept within the agreed bounds.

All three levels of QOS commitment could be supported in the network using a combination of deadline scheduling and priority queuing. In a priority based scheme, each traffic class (deterministic, probabilistic and best effort) is characterised by a given priority. This ensures that deterministic traffic has priority over probabilistic traffic which in turn has priority over best effort traffic. An alternative strategy is possible which allocates fixed network resources e.g. deterministic and probabilistic traffic receive 85% of the bandwidth and best effort only 15%. This guarantees the performance of deterministic and probabilistic communications and avoids starvation of datagram communications, but may lead to lower overall utilisation of bandwidth because statistical gain may not be realised.

Research on the network scheduling problem is reported in, for example, [11] and [10]. A significant amount of work is currently being undertaken in this field.

Apart from active scheduling, ATM networks will require a suitable resource management mechanism which is able to allocate network resources such as bandwidth and end-to-end delay at set-up time. A resource reservation mechanism will allocate network resources at the end systems and at all intermediate switches between the source and destination. A policing mechanism is also required to monitor the user behaviour at the network edge. The network has to exercise flexible resource allocation and congestion control to exploit the potential increase in network efficiency resulting from the use of statistical multiplexing.

4.3.4.3 Example

The final stage in the QOS mapping example is the generation of the following ATM layer QOS specification.

```
QOS commitment level: deterministic
QOS parameters
         peak cell arrival rate: 10 Mbps
         average cell arrival rate: 8 Mbps
         burstiness: 5
         peak duration: 100 ms
Call control parameters
         establishment delay: 4 seconds
         establishment failure prob: 0.05
       (i.e. 95% chance to connect)
         release delay: 4 seconds
Reliability parameters
         BER: 10^-7 with FEC
         CLR: 10^-5 with FEC
         CIR: 10^-3
Delay parameters
         delay: 200 ms
         jitter: 20 ms
```

It is clear from this final stage of the example that the mapping process is non-trivial and further work is necessary to automate this process in practice.

5. Related Work

5.1 Research

There is currently very little literature on the integrated treatment of QOS. One contribution [12] examines the requirements for QOS support in Open System standards and proposed enhancements to the existing OSI RM to support a QOS framework. Several projects in the RACE programme are concerned with QOS, in particular QOSMIC (R.1082). However, RACE restricts itself to a network-level view of QOS and therefore there is little consideration of higher level QOS issues.

In the higher layers a number of research teams has looked at QOS as a transport layer issue. Work at Heidelberg [13] has also investigated the integration of transport QOS and resource management (scheduling) in the operating system.

An important requirement for the generalised QOS-A in non ATM networks is a suitable reservation scheme which is able to set up and guarantee network resources such as bandwidth and end-to-end delay for high performance continuous media communications between network subscribers. This could be achieved using resource reservation protocols based on such as ST-II [7].

Clark, Shenker and Zhang [10] describe an Integrated Service Packet Network which can support three levels of service commitment: *(i) guaranteed service* for real-time applications; *(ii) predicted service* which utilises the measured performance of delays and is targeted towards adaptive or continuous media applications which can compensate for momentary loss of QOS; and *(iii) best effort* datagram service where no QOS guarantees are provided. Also, a unified traffic scheduling mechanism is developed which is based on a combination of weighted fair queuing and static priority algorithms.

5.2 Standardisation

Standards have an important role to play in development of a QOS-A for high performance distributed computing. The current OSI and CCITT standards do not take into account the particular requirements of an integrated QOS model. It can therefore be anticipated that the OSI RM will have to be extended to support the new QOS requirements [1].

In ISO, a New Work Item on QOS has recently been initiated (ISO/IEC JTC1/SC21, and in the UK IST21/-/1/5) which aims to address QOS in a consistent way. This activity will cover QOS very broadly, but has begun by investigating user requirements for QOS and some architectural issues [14]. Lancaster University has provided early input on our QOS-A [15] work into this activity, and the intention is to participate actively in it at both national and international level.

6. Conclusion and Future Work

This paper has argued for a comprehensive architectural framework for QOS support in the light of new applications with varied QOS requirements and new networks able to support QOS configurability. We have presented a draft QOS-A based on our experimental work and have discussed requirements and possible mechanisms for QOS support.

However, much remains to be done. Firstly, the number of QOS dimensions in the architecture can be expanded. Candidates include security and cost dimensions and there are probably others from application domains other than the field of continuous media support which we have been concerned with. Other work remains to be done in the vertical aspects of the architecture. In particular, the specification of QOS at the different layers is a major aspect of further work, as is the process of mapping between the layers. Also required is the definition of a standard framework for QOS negotiation, resource management, monitoring and policing over all the layers. Finally, research must be carried out on QOS protocol support mechanisms. This applies particularly to scheduling which is fundamental at the network layer and is also important at the operating system level where continuous media is concerned.

As the next phase of our research we intend to refine and implement the QOS-A in an experimental configuration consisting of workstations connected via an ATM switch. The workstations will be equipped with audio and video cards and will run our existing transport and orchestration software on a multimedia network interface (MNI) which we have built in a previous project. The workstations will also run a real-time operating system and our multimedia enhanced ANSA application platform to provide a test bed for thread scheduling for continuous media streams. On the ATM side, we also intend to experiment with scheduling mechanisms and to provide a prototype adaptation layer with negotiation, policing and resource management facilities.

Acknowledgement

Part of this work was carried out within the MNI project (funded under the UK SERC Specially Promoted Programme in Integrated Multiservice Communication Networks (grant number GR/F 03097) and co-sponsored by British Telecom Labs), and part within the OSI 95 project (ESPRIT project 5341, funded by the European Commission).

References

[1] Leopold, H., et al, "Distributed Multimedia Communications System Requirements", *OSI95/Deliverable ELIN-1/C/V3,* Alcatel ELIN Research, A-1210 Vienna, Ruthnergasse 1-7, Austria, April 1992.

[2] CCITT, "Draft Recommendations I.*", CCITT Geneva, May 1990.

[3] ISO. "Basic Reference Model of Open Distributed Processing", Working document on structures and functions, ISO/IEC JTC1/SC21 N4022, 10 November 1989.

[4] Coulson, G., G.S. Blair, N. Davies, and A. Macartney. "Extensions to ANSA for Multimedia Computing", Computer Networks and ISDN Systems, 25, 1992.

[5] Campbell, A., Coulson G., Garcia F., and D. Hutchison, "A Continuous Media Transport and Orchestration Service", Presented at ACM SIGCOMM ‘92, Baltimore, Maryland, USA, August 1992.

[6] Shepherd, W.D., D. Hutchison, F. Garcia and G. Coulson. "Protocol Support for Distributed Multimedia Applications." Second International Workshop on Network and Operating System Support for Digital Audio and Video, IBM ENC, Heidelberg, Germany, Nov 18-19 1991.

[7] Topolcic, C. "Experimental Internet Stream Protocol, Version 2 (ST-II)", Internet Request for Comments No. 1190 RFC-1190, October 1990.

[8] APM Ltd. "The ANSA Reference Manual Release 01.01", Architecture Projects Management Ltd., Poseidon House, Castle Park, Cambridge, UK, July 1989.

[9] Govindan, R., and D.P. Anderson. "Scheduling and IPC Mechanisms for Continuous Media." Thirteenth ACM Symposium on Operating Systems Principles, Asilomar Conference Center, Pacific Grove, California, USA, SIGOPS, Vol 25, Pages 68-80.

[10] Clark, D.D., Shenker S., and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", Internal Report, Laboratory for Computer Science, MIT, USA, January 1992.

[11] Zhang, H., and S. Keshav, "Comparison of Rate-Based Service Disciplines" CACM, October 1991, pages 113-121.

[12] Sluman, C., "Quality of Service in Distributed Systems", BSI/IST21/-/1/5:33, British Standards Institute, UK, October 1991.

[13] Hehmann, D.B., R.G. Herrtwich, W. Schulz, T. Schuett, and R. Steinmetz. "Implementing HeiTS: Architecture and Implementation Strategy of the Heidelberg High Speed Transport System" *Second International Workshop on Network and Operating System Support for Digital Audio and Video,* IBM ENC, Heidelberg, Germany, 1991.

[14] ISO, "Quality of Service Framework - Outline", ISO/IEC JTC1/SC21/WG1 N1145, International Standards Organisation, UK, March 1992.

[15] Campbell, A., G. Coulson and D. Hutchison, "A Suggested QOS Architecture for Multimedia Communications", ISO/IEC JTC1/SC21/WG1 N1201, International Standards Organisation, UK, November, 1992.