A Review of Quality of Service Architectures

Cristina Aurrecoechea, Andrew Campbell and Linda Hauw
Center for Telecommunication Research
Columbia University, New York
e-mail: {cris, campbell, linda}@ctr.columbia.edu

ABSTRACT

The integration of distributed multimedia systems support into a communications architecture, including new multiservice networks, is important in realising end-to-end *quality of service (QoS)* guarantees. A key observation is that quality of service provides a unifying theme around which new *QoS architecture* can be constructed. For applications relying on the transfer of multimedia, and in particular continuous media flows, it is essential that quality of service is configurable, predictable and maintainable system-wide, including the end-system devices, communications subsystem and networks. Although researchers have addressed many isolated areas of QoS provision, until recently little attention had been paid to the development of QoS architecture which incorporates quality of service interfaces, and quality of service control and management mechanisms across all architectural layers. The approach taken in this paper is, first, to set out terminology and *elements of a generalised QoS framework* for understanding and discussing quality of service in distributed multimedia systems, second, to review current research in the area of layer specific quality of service control and management and finally, to evaluate a number of QoS architectures that have emerged in the literature recently.

1. Introduction

Deriving effective *quality of service (QoS)* guarantees in distributed multimedia systems is fundamentally an end-to-end issue; that is, from application-to-application. For example, consider the remote access to a sequence of audio and video: in the distributed system platform, quality of service assurances should apply to the complete flow of media; from the remote server, across the network to the point of delivery. This generally requires end-to-end admission testing and resource reservation in the first instance, followed by careful co-ordination of disk and thread scheduling in the end-system, packet/cell scheduling and flow control in the network, and finally active monitoring and maintenance of the delivered quality of service. In other words, in order to meet distributed multimedia application requirements it is essential that quality of service is assured on an end-to-end basis (as illustrated in Figure 1-1). And futhermore, it is also essential that all end-to-end elements of distributed systems architecture work together in unison to achieve the desired application level behaviour.

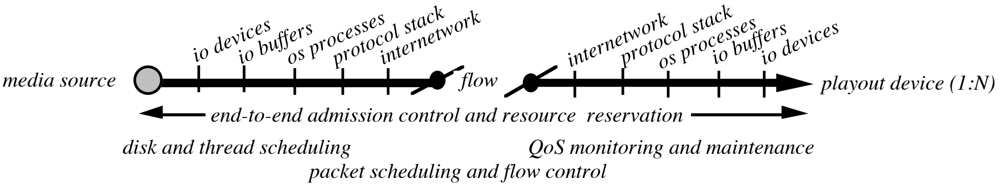

Figure 1-1. End-to-End QoS Scenario

Most of the developments in the provision of quality of service support have occurred in the context of individual architectural layers. Much less progress has been made in addressing the issue of an overall *QoS architecture* for multimedia communications. There has been, however, considerable progress in the area of operating systems, transports and networks support for quality of service. In end-systems, most of the progress has been made in the specific areas of flow synchronisation, communication and scheduling support. In networks, research has focused on providing suitable traffic models and service disciplines, as well as appropriate admission control and resource reservation protocols. Many current network architectures, however, address quality of service from a providers point of view and analyses network performance, failing to comprehensively address the quality of needs of distributed multimedia applications. Until recently there has been little work on quality of service support in distributed systems platforms. What work there is has been mainly in the context of the Open Distributed Processing (ODP).

The current state of QoS provision can be summarized as follows [1]:

- i) *incompleteness*: current interfaces are generally not QoS configurable and provide only a small subset of facilities needed for control and management of multimedia flows;
- ii) *lack of mechanisms to support QoS guarantees*: research is needed in distributed control, monitoring and maintenance QoS mechanisms so that contracted levels of service can guaranteed; and
- iii) *lack of overall framework*: it is necessary to develop an overall architectural framework to build on and reconcile the existing notion of QoS at different systems levels and among different network architectures.

In recognition of the above limitations, a number of research teams have proposed a systems architectural approach to QoS provision for networked multimedia systems; we refer to these models as QoS architectures in this paper. The intention of QoS architecture research is to extend the current systems approach by defining a set of quality of service configurable interfaces that formalize quality of service in the end-system and network providing a framework for the integration of quality of service control and management mechanisms.

The structure of this paper is as follows. We first present, in section 2, a *generalized QoS framework* and terminology for distributed multimedia applications operating over multimedia networks with quality of service guarantees. The QoS framework is based on a set of QoS principles that govern the behavior of QoS architectures. Following this, we review current layer-specific work on quality of service support (in section 3) considering the distributed systems platform layer, operating systems layer, and transport and network layers. In section 4, we evaluate four distinct QoS architectures found in the literature. Then in section 5, we review related work in QoS architecture. Following this we present a short qualitative comparison of QoS architecture surveyed in this paper. Finally, in section 6 we offer some concluding remarks.

2. Elements of a Generalised QoS Framework

In what follows, we describe a set of QoS elements used in building quality of service into distributed multimedia systems. This includes QoS principles which govern the construction and behaviour of a generalised QoS framework, QoS specification which captures application level quality of service requirements, and QoS mechanisms which realise desired end-to-end QoS behaviour.

2.1 QoSPrinciples

Five principles motivate the design of a generalised QoS framework:

- i) *integration principle* states that quality of service must be configurable, predictable and maintainable over all architectural layers to meet end-to-end quality of service [2]. Flows[1] traverse resource modules (e.g., CPU, memory, devices, network, etc.) at each layer from source media devices, down through the source protocol stack, across the network, up through the receiver protocol stack to the playout devices. Each resource module traversed must provide QoS configurability (based on a QoS specification), resource guarantees (provided by QoS control mechanisms) and maintenance of the on-going flows (realised by QoS management mechanisms);
- ii) *separation principle* states that media transfer, control and management are functionally distinct activities in the architecture [3]. The principle states that these tasks should be separated in the architecture; one aspect of separation is the distinction between signalling and media-transfer; flows (which are simplex and isochronous in nature) generally require a wide variety of high bandwidth, low latency, non-assured services with some form of jitter correction; on the other hand, signalling (which is full duplex and asynchronous in nature) generally requires low bandwidth, assured-type services with no jitter constraint;
- iii) *transparency principle* states that applications should be shielded from the complexity of underlying QoS specification and QoS management [4] such as QoS monitoring and maintenance. An important aspect of transparency is the QoS-based API at which desired quality of service levels are stated (see QoS management policy).

[1] The notion of a flow is an important abstraction which underpins the development of QoS frameworks. Flows characterize the production, transmission and eventual consumption of a single media source (viz. audio, video, data) as integrated activities governed by single statements of end-to-end QoS. Flows are simplex in nature and can be either unicast or multicast. Flows generally require end-to-end admission control and resource reservation, and support heterogeneous QoS demands.

The benefit of transparency is three-fold: to reduce the need to embed quality of service functionality in applications; hiding the detail of underlying service specification from the application; and delegating the complexity of handling QoS management activities to the underlying framework;

- iv) *asynchronous resource management principle* guides the division of functionality between architectural modules [3] and pertains to the modeling of control and management mechanisms; it is necessitated by, and is a direct reflection of fundamental time constraints that operate in parallel between activities (e.g., scheduling, flow control, routing, QoS management, etc.) in distributed communications environments; the "state" of the distributed communication system is structured according to these different time scales. The communication system 'operating point' is arrived at via asynchronous algorithms that operate and exchange control data periodically among each other; and
- v) *performance principle* includes a number of widely agreed rules for QoS-driven communications implementation that guides the division of functionality in structuring communication protocols for high performance in accordance with Saltzer's systems design principles [5]; avoidance of multiplexing [6]; recommendations for structuring communications protocols such as application layer framing and integrated layer processing [7], and the use of hardware assists for protocol processing [8] [9].

2.2 QoS Specification

QoS specification is concerned with capturing application level quality of service requirements and management policies. QoS specification is generally different at each system layer and is ultimately used to configure and maintain QoS mechanisms resident at each layer. For example, at the distributed system platform level QoS specification is primarily user-oriented rather than system-oriented. Lower-level considerations such as tightness of synchronisation of multiple related flows, or the rate and burst size of flows, or the details of thread scheduling should all be hidden at this level. QoS specification is therefore declarative in nature; whereby users specify what is required rather than how this is to be achieved by underlying QoS mechanisms. Quality of service specification considers the following:

- *flow synchronisation specification*, which characterises the degree (i.e., tightness) of synchronisation between multiple related flows [10]. For example, simultaneously recorded video perspectives must be played in precise frame by frame synchrony so that relevant features may be simultaneously observed. On the other hand, lip synchronisation in multimedia flows does not need to be absolutely precise when the main information channel is auditory and video is only used to enhance the sense of presence;
- *flow performance specification,* which characterises the user's flow performance requirements [11]; the ability to guarantee traffic throughput rates, delay, jitter and loss rates, is particularly important for multimedia communications. These performance-based metrics are likely to vary from one application to another; to be able to commit necessary end-system and network resources a QoS framework must have prior knowledge of the expected traffic characteristics associated with each flow before resource guarantees can be met;
- *level of service (LoS)*, which specifies the degree of end-to-end resource commitment required (e.g, deterministic [12], predictive [13] and best effort). While the flow specification permits the user to express the required performance metrics in a quantitative manner, level of service allows these requirements to be refined in a qualitative way as to allow a distinction to be made between hard, firm and soft performance guarantees. Level of service expresses a degree of certainty that the QoS levels requested at flow establishment or re-negotiation will actually be honored;
- *QoS management policy*, which captures the degree of quality of service adaptation (continuous or discrete) that the flow can tolerate and scaling actions to be taken in the event of violations in the contracted QoS [14]. By trading-off temporal and spatial quality to available bandwidth, or manipulating the playout time of continuous media in response to variation in delay, audio and video flows can be kept meaningful at the playout device with minimal perceptual distortion. The QoS management policy also extends to include user-level QoS indications which indicate QoS degradation (i.e., QoS violations) and periodic bandwidth, delay, jitter and loss notification (i.e., QoS signals); and
- *Cost of Service (CoS)*, which specifies the cost the user is willing to incur for the level of service; cost of service is very important factor when considering QoS specification. If there is no notion of cost of service involved in QoS specification, there is no reason for the user to select anything other than maximum level of service [15].

2.3 QoS Mechanisms

Quality of service mechanisms (i.e., algorithms) are driven by user supplied QoS specification, resource availability and resource management policy. In resource management, QoS mechanisms are categorized as either static or dynamic in nature: *static resource management* deals with flow establishment and end-to-end QoS re-negotiation phases (which we describe as QoS provision), *dynamic resource management* deals with the media-transfer phase (which we describe as QoS control and management). The distinction between the former and latter, is due to the different time scales on which they operate and, is a direct consequence of the asynchronous resource management principle. Control distinguishes itself from management in that it operates on a faster timescale.

2.3.1 QoS Provision Mechanisms

The generalised QoS provision is comprised of three components:

- i) *resource reservation protocols* arrange for the allocation of suitable end-system and network resources according to the user QoS specification. In doing so, the resource reservation protocol interacts with QoS-based routing to establish a path through the network in the first instance, then, based on QoS mapping and admission control at each local resource module traversed (e.g. CPU, memory, I/O devices, switches, routers, etc.) end-to-end resources are allocated. The end results is QoS control and management mechanisms such as network-level cell scheduler and transport-level flow monitors are configured appropriately;
- ii) *QoS mapping* performs the function of automatic translation between representations of QoS at different system levels (e.g., operating system, transport layer, network, etc.) and thus relieves the user of the necessity of thinking in terms of lower level specification. For example, the transport level QoS specification may express flow requirements in terms of average and peak bandwidth, jitter, loss and delay constraints. For admission testing and resource allocation purposes this representation must be translated to something more meaningful to the end-system scheduler. As illustrated below, QoS mapping derives the period, quantum (i.e., unit of work), and schedule and deadlines times of the threads associated with transport level flows [16]; and

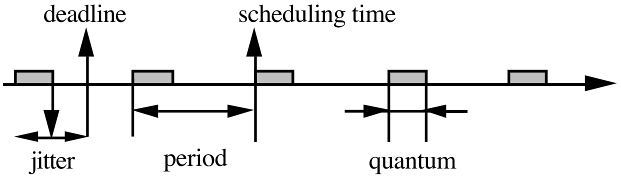

Figure 2-1. QoS Mapping Scenario

- iii) *admission testing* and resource reservation are tightly coupled. Once admission testing has been successfully completed on a particular resource module, local resources are reserved immediately and then committed later if the end-to-end admission control test (i.e., accumulation of hop by hop tests) is successful. Admission control is responsible for comparing the resource requirement arising from the QoS levels requested against the available resources in the system. The decision whether a new request can be accommodated depends on not only on resource availability but also on resource management policies.

2.3.2 QoS Control Mechanisms

QoS control mechanisms operate on timescales close to media transfer speeds. They provide real-time traffic control of flows base on requested levels of QoS established during the QoS provision phase. This is achieved by providing suitable traffic control mechanisms and arranging for time-constrained buffer management and communication protocol operation. The fundamental traffic control building blocks include:

- *flow shaping* regulates flows based on user supplied flow performance specifications. Flow shaping can be based on a simple fixed rate (i.e., peak rate) or some form of statistical representation (i.e., sustainable rate and burstiness). The benefit of shaping traffic is that it allows the QoS framework to commit sufficient end-to-end resources and to configure the flow scheduler to regulate traffic through the end-systems and network.

It has been mathematically proven that the combination of traffic shaping at the edge of the network and scheduling in the network can provide hard performance guarantees. Parekh [17] has shown that if a source is shaped by a token bucket with leaky bucket rate control and scheduled based on weighted fair queueing service discipline [18], it is possible to achieve strong guarantees on delay for that flow;

- *flow scheduling* manages the forwarding of flows in the end-system (chunks of data based on application layer framing) [19][20][21] and network (packets and/or cells) in an integrated manner [22]. Flows are generally scheduled independently in the end-systems but may be aggregated and scheduled in unison in the network. This is dependent of the level of service and the scheduling scheme adopted;
- *flow policing* can be viewed as the duality of monitoring: the latter - usually associated with QoS management - observes whether QoS contracted by a provider is being maintained whereas the former observes whether the QoS contracted by a user is being adhered to. Policing is often only appropriate where administrative and charging boundaries are being crossed, for example, at a user-to-network interface [23]. A good flow shaping scheme at the source allows the policing mechanism to easily detect misbehaving flows. The action taken by the policing function can range from accepting violations and merely notifying the user, through to shaping the incoming traffic to an acceptable QoS level. We consider that policing flows in the end-system or network should be a function of the end-system or network level scheduling QoS mechanism;
- *flow control* includes both open-loop and closed loop schemes: open loop flow control, which is used widely in telephony allows the sender to inject data into the network at the agreed levels given resources have been allocated in advance; closed loop flow control requires the sender to adjust its rate based on feed-back from the receiver or network [24]. Applications using closed loop flow control based protocols must be able to adapt to fluctuations in the available resources. Fortunately, many multimedia applications are adaptive [25][26] and can operate in such environments. Alternatively, multimedia applications which can not adjust to changes in the delivered QoS are more suited to open loop schemes where bandwidth, delay and loss can be deterministically guaranteed for the duration of the session; and
- *flow synchronisation* is required to control the event ordering and precise timings of multimedia interactions. Lip-sync being the most commonly cited form of multimedia synchronisation (synchronisation of video and audio flows at a playout device); other synchronisation scenarios reported include: event synchronisation with and without user interaction, continuous synchronisation other than lip-sync, continuous synchronisation for disparate sources and sinks. All place fundamental QoS requirements on flow synchronisation protocols [27]. Dynamic QoS management associated with flow synchronisation is mainly concerned with the ‘tightness’ of synchronisation between flows.

2.3.3 QoS Management Mechanisms

To maintain agreed levels of QoS it is often not sufficient to just commit resources; in addition, QoS management is frequently required to ensure that the contracted QoS is sustained. QoS management of flows is functionally similar to QoS control. However, it operates on a slower time scale; that is, over longer monitoring and control intervals [28]. QoS management mechanism include:

- *QoS monitoring* allows each level of the system to track the ongoing QoS levels achieved by the lower layer. It often plays an integral part in a QoS maintenance feedback loop which maintains the quality of service being achieved by the monitored resource modules. Monitoring algorithms operate over different timescales. For example, they can run as part of the scheduler (as a QoS control mechanism) to measure individual performance of on-going flows. In this case measured statistics can be used to control packet scheduling and for admission control. Alternatively they can operate as part of a transport level feedback mechanism [49];
- *QoS maintenance* compares the monitored quality of service against the expected performance and then exerts QoS tuning (i.e., fine or coarse grain resource adjustments) on resource modules to sustain the delivered QoS. Fine grain resource adjustment counters QoS degradation by adjusting local resource modules (e.g., loss via the buffer manager, queueing delays via the flow scheduler and throughput via the flow regulator [2]);
- *QoS degradation* issues a QoS indication to the user when it determines that the lower layers have failed to maintain the QoS of the flow and nothing further can be done by the QoS maintenance mechanism. In response to such an indication the user can choose either to adapt to the available level of QoS or scale to a reduced level of service (i.e., end-to-end renegotiation);

- *QoS signal* allows the user to specify the interval over which one or more QoS parameter (delay, jitter, bandwidth, loss, synchronisation) can be monitored and the user informed of the delivered performance. Both single and multiple quality of service signals can be selected depending QoS management policy; and
- *QoS scalability* comprises *QoS filtering* (which manipulates flows as they progress through the communications system) and *QoS adaptation* (which scales flows at the end-systems only) mechanisms. Many continuous media applications exhibit robustness in adapting to fluctuations in end-to-end quality of service. Based on the user supplied QoS management policy, QoS adaptation in the end-systems can take remedial actions to scale flows appropriately. Resolving heterogeneous quality of service issues is a particularly acute problem in the case of multicast flows. Here individual receivers may have differing capabilities to consume audiovisual flows; QoS filtering helps to bridge this heterogeneity gap while simultaneously meeting individual receivers' quality of service requirements.

3. Layer-specific QoS

In this section we selectively review layer-specific quality of service research considering the distributed systems platform, operating system, and transport and network layers in turn below; see [29] [30] for a more complete survey.

3.1 Distributed Systems Platform

There has been considerable research in the area of distributed systems platform over the past ten years [31]. Until recently, however, there has been very little work on quality of service support in such platforms. With the emergence of distributed multimedia applications, however, quality of service has become a major issue in distributed systems research. In a distributed system, there are three areas where quality of service is applicable: i) message passing services, which allow a programmer to explicitly send a message between two or more processes in a distributed system; ii) remote invocation, which allows operations in a server process to be invoked by a client process [32] [33]; and iii) stream services, which are connections that support the transmission of continuous media flows [34]. A number of experimental QoS-driven distributed systems platforms are now beginning to emerge. Researchers at Lancaster University have developed an extended version of ANSAware [32] featuring bounded invocations and QoS-controlled streams [35]. Similar work has also been undertaken at Cambridge University [36]. More recently, research on quality of service has centered on ODP standardization [34]. Ongoing research at CNET [37], and BBN and Rome Labs [38] are developing new languages to specify QoS for both operational and stream interface. The CNET work uses QoS logic statements in the language to generate quality of service monitors. The BBN and Rome Lab research promotes object level QoS specification (i.e., methods per second) and not at the communication level (i.e.,bits per second). Both approaches allow quality of service to be negotiated, measured and enforced. For full details on the state of the art in distributed systems support for quality of service see [1].

3.2 Operating Systems

There has been considerable progress in operating systems support for multimedia with most progress having been made in the specific areas of communication protocols [40], scheduling [20] and end-system architecture [39]. There has been considerably less work on the integration of the various components into an overall operating systems [16]. Communication protocol implementation involves predictability issues such as the need for correct scheduling of protocol activities and efficiency issues such as minimization of data copying, system calls, interrupt handling and context scheduling, an avoidance of multiplexing, the use of hardware assists for protocol processing and the importance of executing protocol code in a schedulable process rather that as a interrupt service routine. Much of the work has looked to maintain a level of compatibility with the de facto UNIX interface. Two main approaches can be identified: i) modifying existing UNIX implementations, and ii) completely re-implementing UNIX. In the first approach, alterations are made to the existing UNIX kernel to provide more predictable behaviour. For example, a range of projects is currently under way at SUN Mircosystems in this area. Their proposal is for time-driven resource management [41] which allows applications to signal their likely forthcoming resource requirements in terms of QoS parameters such as quantity deadline and priority. The second approach is in terms of the mirco-kernel model. Examples of mirco-kernels capable of supporting UNIX interfaces are Chorus, Mach and Amoeba. Work has been undertaken at CWI, Amsterdam to support continuous media in an Amoeba-based UNIX environment [42]. Other significant work is being carried out using Mach [21], Chorus[16], Peagus [43] as the basis of a distributed system with end-to-end QoS support. For full details on the state of the art in operating systems support for quality of service see [1].

3.3 Transport Layer

A large number of research teams have investigated the provision of quality of service at the transport layer. Early work specifically addressed the provision of rate based protocols over high speed networks, e.g., XTP [8] and NetBlt [44]. More recently protocols have emerged which are designed specifically to meet the needs of continuous media. The recent Esprit OSI 95 project has proposed an enhanced transport service and protocol collectively described as TPX [45]. TPX provides support for connection-oriented services with sequenced delivery, QoS configurable and renegotiable QoS, and error notification. The enhanced connection-oriented service takes QoS parameters relating to throughput, delay, delay jitter, error selection policy and relative priority. Three transport quality of service semantics in addition to "best effort" are proposed for this service: compulsory, threshold and maximal QoS. The Tenet Group at the University of California at Berkeley have developed CMTP[46] which operates on top of RTIP [83] and provides sequenced and periodic delivery of continuous media samples with QoS control over throughput, delays and error bounds. Notification of all undelivered and/or corrupted data can be provided if the client selects this option. The HeiTS project [47] at IBM Heidelberg have developed a transport system which has concentrated on the integration of transport QoS and resource management (primarily CPU scheduling). HeiTS puts considerable emphasis on an optimized buffer pool which minimizes copying and also allows efficient data transfer between local devices. One significant work has come from Schulzrinne, Casener and Van Jacobson who have developed RTP [48] for the Internet suite of multimedia tools [26]. Other work [49] reports on the development of a continuous media transport and orchestration service. For a full review of the state of the art in transport protocols and services see [40] [50].

3.4 Network Layer

The subject of providing quality of service guarantees in integrated service networks has been widely covered in the literature [51]. The multimedia networking community has developed sophisticated traffic models, control and management architectures for multimedia communications. Extensive work has considered flow specification, flow admission control, resource reservation, traffic shaping and queue management schemes. For researchers working on multimedia networking, the notion of QoS is a fundamental one of providing performance bounds while exploiting statistical multiplexing of bursty sources to efficiently utilise bandwidth. Kurose [52] provides a good categorisation of the different approaches used in providing QoS guarantees found in the literature: (i) *tightly controlled approach*, which is based on non-work conserving multiplexing service (queueing) disciplines (e.g., stop-and-go [53] and TENET's EDD [12]), preserves the traffic shape guaranteeing delivered flow characteristics are the same as the source; (ii) *approximate approach*, which as its names suggests is based on simple characterisation of the source model (e.g., equivalent capacity [54]) can provide approximate guarantees using simple service disciplines such as FIFO taking advantage of statistical multiplexing gain ; (iii) *bounding approach*, which takes into any account distortion of the flow as traverses work-conserving multiplexers (e.g., packetised generalised processor sharing [17] and weighted fair queueing [18]) resulting in mathematically provable performance bounds for statistical and deterministic service guarantees [55]; and finally (iv) *observation-based approach*, which uses measured behaviour (e.g., COMET's approach [56] and Clark's predictive service [13]) of the aggregate traffic and the user suppled flow specification when making admission decisions.

The work on an integrated services Internet [57] is a significant contribution to providing QoS guarantees on a perflow basis. The integrated service model includes four components: (i) *packet scheduler*, which is based on the CSZ scheduler [13] and Class Base Queueing (CBQ) [58], forwards packet streams using a set of queues and timers; (ii) *classifier* maps each incoming packet into a set of QoS class (iii) *admission controller* implements the decision control algorithm to determine whether a new flow can be admitted or denied; (iv) *reservation setup protocol* is necessary to create and maintain flow-specific state in the end-systems and in routers along the path of the flow. There have been a number of significant contributions to reservation protocols in communication networks which have emerged over the past few years: ST-II [59] and SRP [86], and more recently RSVP [60], RCAP [61] and HieRAT [62] and UNI 3.0 [23]. For a full review of the state of the art in network support for QoS see [51] [52].

4. QoS Architectures

Until recently research in providing QoS guarantees has mainly been focussed on network oriented traffic models and service scheduling disciplines. These guarantees are not, however, end-to-end in nature. Rather they preserve QoS guarantees between network access point that end-systems are attached to [63]. Likewise work on QoS-driven end-system architecture needs to be integrated with network configurable QoS services and protocols to meet application-to-application QoS guarantees. In recognition of this, researchers have recently proposed new communication architectures which address this limitation. In this section we review four distinct approaches which are born out of the Telecommunications and Computer Communications communities. These are the *Extended Integrated Reference Model* (developed at Columbia University) and *TINA QoS Framework* (developed by the TINA Consortium) which have emerged from the telecommunication community; and the *Quality of Service Architecture* (developed at Lancaster University) and *MASI End-to-End Architecture* (developed at University Pierre et Marie Curie) which have evolved from work on computer-to-computer communications.

4.1 The Columbia Extended Integrated Reference Model

The COMET group at Columbia University (New York) are developing an Extended Integrated Reference Model (XRM) [64] as a modeling framework for control and management of telecommunications multimedia networks (which comprise of *multimedia computing platforms* and *broadband networks*). The COMET group argues that the foundations for operability (i.e., control and management) of multimedia computing and networking devices are equivalent; that is, both classes of devices can be modeled as producers, consumers and processors of media.The only difference is the overall goal that a group of devices has set to achieve in the network or end-system. COMET organizes the XRM into five distinct planes as illustrated in Figure 4-1:

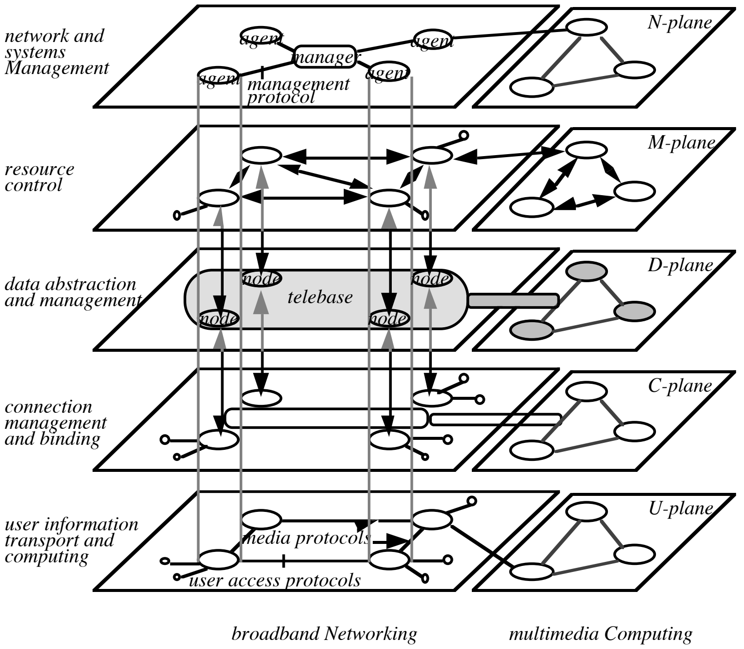

Figure 4-1. XRM Schematic

- *management function*, which resides in the network management plane (N-plane), covers the OSI functional areas of network and system management;
- *traffic control function*, which comprises the resource control (M-plane) and connection management and control (C-plane) planes. Resource control constitutes cell scheduling, call admission, call routing in the network and, process scheduling, memory management, routing (when applicable [39]), admission control and flow control in the end-systems;
- *information transport function*, which is located in the user transport plane (U-plane), models the media protocols and entities for the transport of user information in both the network and the end- systems; and
- *telebase*, which resides in the data abstraction and management plane (D-plane), collectively represents the information, data, abstractions existing in the network and end-systems. The telebase implements the principles of data sharing (via asynchronous resource management) among all other XRM planes.

The subdivision of XRM into these different planes is motivated by a number of QoS principles: the separation and layering principles [3], and the principle of asynchronous resource management [3]. The subdivision between the management and traffic control functions, on one hand, and the information transport functions on the other, is based on the principle of separation between control and communication. The separation between management and traffic control is due to the different timescales on which these planes operate; this is in turn motivated by the asynchronous resource management principle.

The XRM is built on sound theoretical work of guaranteeing QoS requirements in ATM networks and end-systems populated with multimedia devices. General concepts for characterising the capacity of network [65] and end-system [66] devices (e.g., disks, switches, etc.) has been developed. At the network layer, XRM characterises the capacity region of an ATM multiplexer with QoS guarantees as a *schedulable region*. Network resources such as switching bandwidth and link capacity are allocated based on four cell-level traffic classes (class I, II, III, and C) for circuit emulation, voice and video, data, and network management respectively. A traffic class is characterised by its statistical properties and QoS requirements. Typically QoS requirements reflect cell loss and delay constraints. In order to efficiently satisfy the QOS requirements of cell level, scheduling and buffer management algorithms dynamically allocate communication bandwidth and buffer space appropriately.

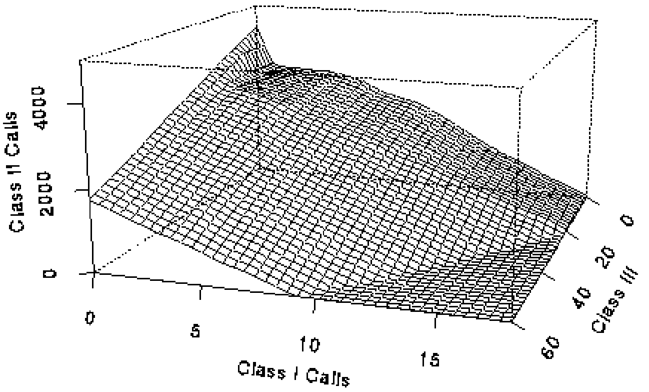

Figure 4-2. The Schedulable Region of a Multiplexer with Three Traffic Classes.

The schedulable region represents the multidimensional capacity of the multiplexer; its dimensionality depends on the number of traffic classes and represents the stability region. The schedulable region is a resource abstraction that allows a separation of times scales: the time scales of cells and the time scale of call arrivals and departures. In [65] it is shown how separation of time scales is an appropriate tool for resolving admission control decisions. Based on a calculus of schedulable regions, the QoS in the network can be guaranteed. The three traffic classes in Figure 4-2 correspond to video, voice and data flows. Class I traffic is characterised by a frame duration of 62,5 ms and a peak rate of 10 Mbps, Class II traffic is modelled as an on-off source with constant arrivals with and exponentially distributed active period and 64 Kbps peak rate, and Class III traffic is modeled as a Poisson source with 1 Mbps average rate. The surface depicted in Figure 2 delimits the capacity region of the multiplexer. Any combination in the number of calls (i.e., active flows) below this surface has its QoS guaranteed.

XRM models the end-system architecture as multiprocessor based multimedia workstation, comprising the following multimedia devices: (i) an *audio and video unit*, which is responsible for multimedia processing, and supports media processing tasks in a deterministic manner, and runs on a dedicated processor(s); (ii) *input/output subsystem* is similarly modeled, separately through a disk storage unit, and is also run on a separate processor(s); (iii) the main processor unit runs the system tasks, both to increase speed and to remove external interrupts, as well as the other operating system overhead associated with application tasks. In the end-system, flow requirements are modeled through service class specifications with QoS constraints. For example, in the audio video unit the service class specification is in terms of JPEG, MPEG-I, MPGE-II video and CD audio quality flows with QoS guarantees. Quality of service for these classes is specified by a set of frame delay and loss constraints.The methodology of characterising network resources is extended to the end-system to represent the capacity of multimedia devices. Using the concept of a *multimedia capacity region* the problem of scheduling flows in the end-system becomes identical to the real-time bin packing exercise of the network layer. One significant difference between the schedulable region and the multimedia capacity region is the number of classes supported. The number of service classes at the user level is expected to far exceed the number of traffic classes at the multiplexer. A number of service classes, however, can be mapped onto a single traffic class of the multiplexer, and therefore, supporting a large number of service classes will not require an increase in the number of traffic classes.

The implementation of XRM including key resource abstractions such as the schedulable and multimedia capacity region is currently being realised as part of a *binding architecture* [67]. The binding architecture achieves seamless binding between networking and multimedia devices. The building blocks of the architecture consist of a set of interfaces, methods and primitives. The former abstract the functionalities of multimedia networking devices and are organised into a *binding interface base (BIB)*. The methods and primitives are invoked for implementing binding applications. Communication between the interfaces of the architecture is supported by OMG's CORBA [70]. Binding requirements arise in each of the planes of the XRM. Dynamic binding requirements, however, are particularly demanding in the C and M planes of the XRM. The binding architecture resides in the M, D and C-planes of the XRM. Specifically, the binding interface base resides in the D-plane and the binding algorithms execute from within the M and C-planes. The binding architecture represents a software environment on top of which binding applications execute. Examples of binding applications arise in connection set up for broadband networks, distributed systems implementing flow synchronisation protocols , resource allocation protocol such as those intended for the Internet [60], multimedia computing platforms, etc. New binding applications can be added without changing the underlying binding architecture.

4.2 The Lancaster Quality of Service Architecture

Over the last three years the QoS-A Project at Lancaster University [2] has been developing a Quality of Service Architecture (QoS-A) in co-operation with ATM switch manufacture GDC (formally Netcomm Ltd). The QoS-A promotes the idea of integrated QoS, spanning the end-systems and network, and takes the support of audio and video flows as its primary design goal. The QoS-A is a layered architecture of services and mechanisms for quality of service (QoS) management and control of continuous media flows in multiservice networks. The architecture incorporates the following key notions: flows characterise the production, transmission and eventual consumption of single media streams (both unicast and multicast) with associated QoS; service contracts are binding agreements of QoS levels between users and providers; and flow management provides for the monitoring and maintenance of the contracted QoS levels. The realisation of the flow concept demands active QoS management and tight integration between device management, thread scheduling, communications protocols and networks. The QoS-A is based on a set of principles that govern the realisation of end-to-end QoS in a distributed systems environment: the integration, separation, transparency and performance principles.

In functional terms, the QoS-A (as illustrated Figure 4.2) is composed of a number of layers and planes. The upper layer consists of a distributed applications platform augmented with services to provide multimedia communications and QoS specification in an object-based environment [16]. Below the platform level is an orchestration layer which provides jitter correction and multimedia synchronisation services across multiple related application flows [Campbell,92]. Supporting this is a transport layer which contains a range of QoS configurable services and mechanisms. Below this, an Internetworking layer and lower layers form the basis for end-to-end QoS support. For full details on the QoS-A see [68].

QoS management is realised in three vertical planes in the QoS-A. The protocol plane, which consists of distinct user and control sub-planes, is motivated by the principle of separation. QoS-A uses separate protocol profiles for the control and media components of flows because of the essentially different QoS requirements of control and data. The QoS maintenance plane contains a number of layer specific QoS managers. These are each responsible for the fine grained monitoring and maintenance of their associated protocol entities. For example at the orchestration layer, the QoS manager is interested in the tightness of synchronisation between multiple related flows. In contrast, the transport QoS manager is concerned with intra-flow QoS such as bandwidth, loss, jitter and delay. Based on flow monitoring information and a user supplied service contract, QoS managers maintain the level of QoS in the managed flow by means of fine grained resource tuning strategies. The final QoS-A plane pertains to flow management, which is responsible for flow establishment (including end-to-end admission control, QoS based routing and resource reservation), QoS mapping (which translates QoS representations between layers) and QoS scaling (which constitutes QoS filtering and adaptation for coarse grained QoS maintenance control).

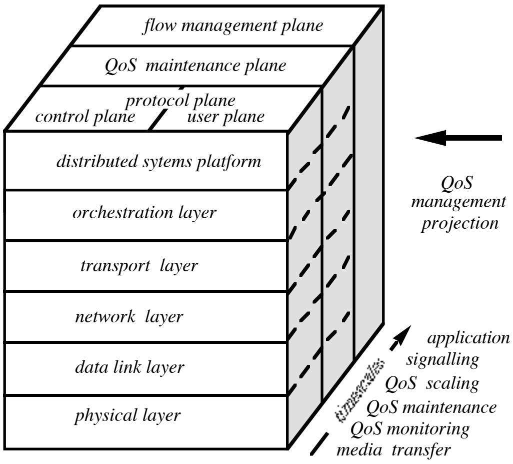

Figure 4-3. QoS-A schematic

Recent work on the QoS-A has concentrated on realising the architecture in an environment comprising an enhanced Chorus micro-kernel [16], and an enhanced multimedia transport service and protocol [68] in the local ATM environment. While the model is end-to-end in design, the main contribution of the work is end-system architecture. This includes a *multimedia enhanced transport system (METS)*, transport service contract and associated operating systems support. At the transport layer, the support of QoS is dependent on interactions between the transport protocol, transport QoS manager, the flow management plane and the network layer. The transport service contract subsumes the well accepted QoS parameters of jitter, loss, delay and throughput, but also allows the QoS specification of a wider range of options. These are characterised in terms of the following six contractual clauses :

- i) *flow specification* characterises the user's quantitative traffic performance requirements in terms of token bucket characterisation of throughput, jitter, delay and loss, and media characterisation in terms of a flow-id and media type;
- ii) *QoS commitment* specifies the degree of resource commitment required from the lower layers, provides three classes: best effort, adaptive [14] and deterministic;
- iii) *QoS scaling policy* identifies the type QoS adaptation [101], QoS filtering [99] in addition to actions to be taken in the event quality of service violations in the contracted service;
- iv) *QoS maintenance* selects the degree of monitoring and active QoS maintenance required;
- v) *resource reservation* provides either on-demand, fast reservation or advanced reservation services; and finally
- vi) *cost,* which specifies the cost the user is willing to incur for the service requested.

The transport protocol and transport-level QoS manager are tightly coupled to operate in the same time domain. In essence, the transport protocol monitors a flow's on-going performance and the transport QoS manager maintains it. The transport protocol's monitoring mechanism is able to build up a statistical representation of the end-to-end QoS using the performance data supplied in the transport control messages. The resulting flow statistics represents the actual end-to-end QoS experienced by the receivers. The transport QoS manager uses this information and a user supplied flow spec for fine grained QoS tuning. The flow management plane is responsible for a number of static and dynamic QoS control and management functions. The major functions consist of the provision of network signalling infrastructure, end-to-end admission control, and QoS scaling for course grained QoS management based on a user supplied QoS scaling policy [14]. The flow management plane also performs other management functions such as the mapping of QoS representation between layers, the support of the on demand, fast and advanced reservation services. For full details of QoS mapping, end-to-end admission control testing and resource reservation in the context of the Lancaster Chorus and ATM environment see [16].

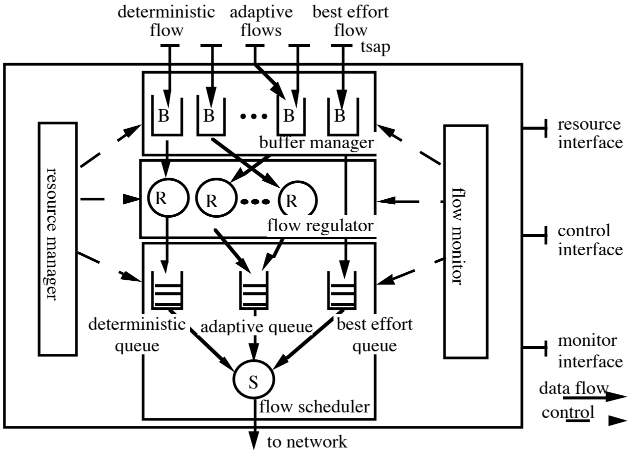

Figure 4-4. Transport-level QoS Mechanisms and Interfaces

In [14] the previous work on a QoS-A is extended by populating the QoS management planes of the architecture with a framework for the control and management of multi-layer coded flows operating in heterogeneous multimedia networking and multicast environments. Two key techniques are proposed: an end-to-end rate shaping scheme which adapts the rate of MPEG-coded flows to the available network resources while minimising the distortion observed at the receiver, and an *adaptive network service*, which offers “hard” guarantees to the base layer of multi-layer coded flows, and “fairness” guarantees to the enhancement layers based on a bandwidth allocation technique called *weighted fair sharing*.

4.3 The TINA Quality of Service Framework

TINA architectural concepts are grouped into four functional domains: Computing Architecture, Service Architecture, Network Architecture and Management Architecture [69]. The TINA approach considers the telecommunications software as a large, distributed software system and applies to it distributed computing and object oriented design techniques. The TINA Computing Architecture, which is largely based on the RM-ODP [34] and influenced by work of the OMG [70], provides the basis for interoperability and reuse of distributed telecommunication software. The TINA QoS Framework [71] describes a framework for specifying QoS aspects of distributed telecommunications within the context of the Computing Architecture. The QoS framework addresses the computational and engineering viewpoints of distributed telecommunications applications. Figure 4-5 illustrates the structure of the telecommunications software in the TINA Computing Architecture. It is governed by the separation between telecommunication applications and *Distributed Processing Environment (DPE)* in the first instance; that is multimedia services offered by a provider utilise the DPE and underlying computing and communications capabilities. These underlying capabilities correspond to operating system functions that are characterised by distinct native configurations. A TINA node comprises a DPE kernel, a Native Computing and Communication Environment (NCCE) and a hardware platform.

The TINA QoS framework is partly based on work in the literature: ANSA QoS Framework [74] and CNET Framework [37]. In the computational viewpoint, QoS parameters required to provide guarantees to objects are stated declaratively as *service attributes*. In the engineering model, QoS mechanisms employed by resource managers are considered. By stating QoS requirements declarative, applications are relieved of the burden of coping with complex resource management mechanisms needed for ensuring QoS guarantees; this is motivated by the principle of QoS transparency.

Computational specification describes applications in terms of computational entities (i.e., objects) that interact with each other. Objects interact via operational interfaces (which correspond to client-server interactions) and stream interfaces (which represents a set of communication end-points as producing or consuming continuous media). Computational objects can support multiple operational and stream interfaces. The TINA QoS Framework supports three types of QoS specification at the object and interface level:

- i) *object QoS specification* details any distinction between the offered and expected quality of service of an object. The quality of service categories currently being considered at this level include availability, security, performance (in terms of response time) and reliability;
- ii) *operational QoS interface specification* focusses on timeliness and availability of quality of service categories: availability is concerned with maximising the likelihood that a service provided is available when requested and, timeliness is concerned with timing constraints of operational interactions;
- iii) *stream QoS interfaces specification* include stream flow signatures and synchronisation constraints on stream flows. The quality of service categories currently being considered at the stream level include throughput, delay, jitter and error rate.

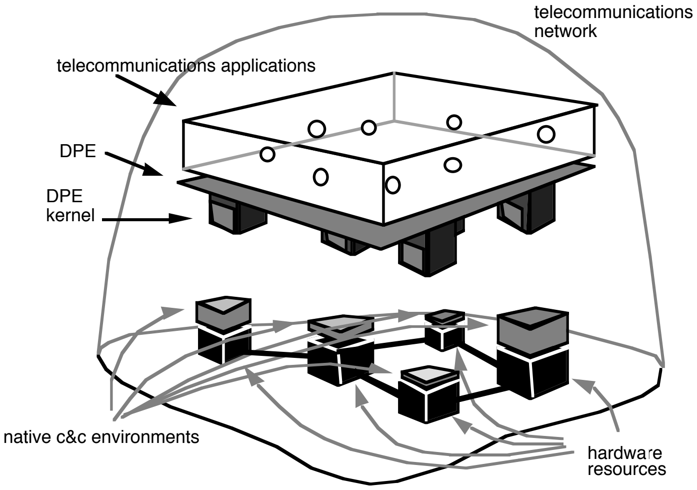

Figure 4-5. TINA-C Schematic

A computational specification language has been developed by the TINA consortium: TINA-ODL, is an extension of OMG-IDL[70] for describing computational objects and their operational and stream interfaces. TINA-ODL provides a service attribute construct to capture the QoS specification of quality of service constraints. In related work [72] an *environmental contract* is introduced. It allows the applications to specify the computational objects and related interfaces in a contract which is a binding agreement between user and service provider. Three levels of QoS are described: deterministic, statistical reliable and best effort. Each quality of service parameter specified in the environmental contract may have a different level of service attributed to it.

The concept of “binding” is used to address the QoS of an interactive session involving multiple computational objects. The binding of computational interfaces is mapped down in the engineering viewpoint as a “channel”. A channel includes three types of functionality: stub, binder and protocol adapter. Figure 4-6 illustrates the computational and corresponding engineering view of a set of objects interacting. In the engineering viewpoint, the objects are distributed in different nodes. The TINA-DPE kernel running in each node is enhanced to offer applications QoS support. Application level QoS requirements are mapped down to services offered DPE kernel and the underlying NCCE. Mechanisms for reporting violations in the contracted quality of service guarantees is provided. Quality of service is considered to be either static (where the service contract is non renegotiable) or dynamic (where the service contract is open to renegotiation by either the DPE kernel or the application). The engineering viewpoint is concerned with which kind of support is required from the environment for realising QoS guarantees. Provision of QoS guarantees is intimately related to static and dynamic resource management of the different kind of resource involved. Hence the engineering viewpoint is interested in identifying the different resource managers involved in the provision of QoS, the resource domain under the control of each resource manager, and how the various resource manager interact and co-operate in the provision of end-to-end QoS.

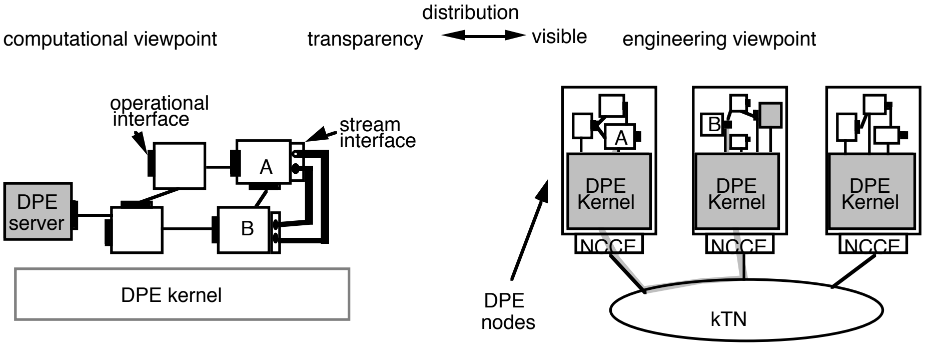

Figure 4-6. Computational and Engineering Viewpoints

Work on the TINA QoS Framework is still in its early stage of development. The approach taken is encouraging even though much work remains. It is important that the work on quality of service in the Computing Architecture is coordinated with on-going work in the rest of the TINA Architectures; for example, in collaboration with the Service Architecture initiative. Furthermore, the DPE nodes illustrated in Figure 4-6 are interconnected to a kernel Transport Network (kTN) [73]. The quality of service provided by the TDPE infrastructure to the computational interactions implicitly rely on the service offered by the NCCE and the kTN combined. This calls for a strong coordination between TINA Computing and Network Architectures, however; in addition, quality of service management activities call for a coordination between the Management Architecture and all other TINA architectural components in turn.

4.4 The MASI End-to-End Architecture

The CESAME Project [75] at MASI Laboratoire, Université Pierre et Marie Curie, is developing an architecture for multimedia communications which takes end-to-end QoS support as it main objective. The MASI architecture provides a framework which offers a generic QoS framework to specify and implement the required QoS requirements of distributed multimedia applications operating over ATM-based networks. The CESAME Project considers end-to-end resource management which span the host operating system, host communication subsytem and ATM networks. The research is motivated by i) the need to map QoS requirements from the ODP layer to specific resource modules in a clean and efficient manner; ii) resolving multimedia synchronisation needs of multiple related ODP streams [34]; and providing suitable communication protocol support for multimedia services.

The MASI architecture addresses the multi-layer, multi-service QoS problem in comprehensive way. Concrete interfaces, mechanisms and services are defined [76]. The architecture comprises of a number of layers (which loosely follow the OSI reference model) and planes (which realise a number of QoS principles) as illustrated in Figure 4-7; these layers include:

- i) *application level*, which refers to an ODP platform which provides QoS conscious services to distributed multimedia applications; see [75] for full details of the QoS specification and support environments;
- ii) *synchronisation layer*, which includes intra flow synchronisation and inter-flow synchronisation between multiple related flows; see [78] for full details of the synchronisation control and management functions; and
- iii) *communication level*, which subsumes the ATM, AAL and transport service and protocol; see [76] for full details and related work at the University of Technology (UTS), Sydney [100].

MASI takes an object-oriented viewpoint, based in part on the RM-ODP [34] approach, for quality of service support of distributed multimedia applications. A number of functions which span the multi-layered architecture are recognised as fundamental in resolving end-to-end resource management issues. These planes comprise:

- *QoS management*, which is the central arbitrator of end-to-end QoS, is comprised of layer specific QoS managers that negotiate resources with peer QoS managers and maintains the internal state associated application specific QoS;
- *connection management*, which manages multimedia session establishment based on a user supplied profile, is made up of layer specific connection managers that *bind multimedia processing units* (MPUs) at each layer in order to meet end-to-end connectivity; and
- *resource management,* which is responsible for host operating systems and communication subsystem resource, performs both admission testing and resource reservation at every level in the end-system.

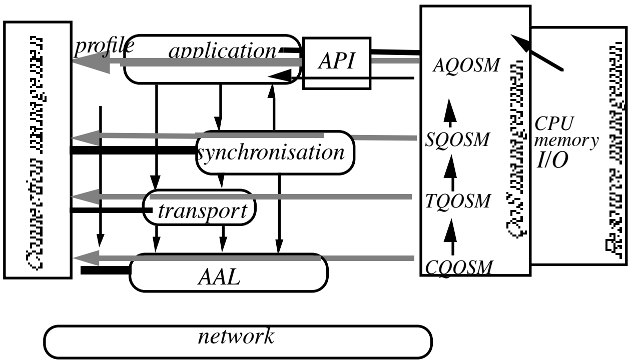

Figure 4-7. MASI Schematic

The Application QoS Manager (AQOSM) translates application requests for multimedia flows into a set of QoS, services and protocol requirements . MASI QoS mapping is based the concept of an application-level *QoS profile.* For each flow, the AQOSM derives suitable profiles.QoS profiles are used in selection of protocol functionality and as a basis for to determine flow specifications used by the communication subsystem. The AQOSM selects desired and minimum values for performance parameters encapsulated in a flow specification [76]. And, furthermore, the AQOSM selects appropriate communication and synchronisation protocol libraries based on the QoS profile template; see [76] for full specification of the profile and the method of protocol selection. A significant function of QoS management is to monitor layer specific QoS and report any QoS violations of the contracted profile directly to the applications. Other QoS violations fielded by QoS management as opposed to being generated by it, include, indications from the resource management plane during the negotiation phase. In this instance the resource management function indicates the level of service of provided to on-going flows; that is, either at the desired or minimum levels.

The MASI architecture focuses on end-system resource management: CPU scheduling, memory and I/O management. Network level admission testing and reservation are left for future work. The CPU scheduling scheme adopted by the CESAME team is based on rate monotonic scheduling (RM) policy [79]. In this instance, the resource management plane actively measures the CPU usage and periodically informs the CPU scheduler of the utilisation. This is then used to accept or deny new flows in the end-system. A novel aspect of the MASI work is the use of application, system and communication libraries which are registered with known attributes in *QoS MIB*. Users' QoS requirements captured in the QoS profile are used as a basis for the dynamic section of the appropriates system functions. Selection of libraries is achieved at flow establishment time, QoS renegotiation time and importantly, dynamically by the resource manager to optimise or change the level of service available to a flow. The construction of profiles through re-usable software modules have been shown to be viable in the CESAME project [75].

5. Related Work

On early contribution to the field of QoS-driven architecture is the *OSI QoS Framework* [80] which concentrates primarily on quality of service support for OSI communications [81]. The OSI framework broadly defines terminology and concepts for QoS and provides a model which identifies objects of interest to QoS in open system standards. The QoS associated with objects and their interactions is described through the definition of a set of QoS characteristics. The key QoS framework concepts include:

- *QoS requirements*, which are realized through QoS management and maintenance entities;
- *QoS characteristics*, which are a description of the fundamental aspects of QoS that have to be managed;
- *QoS categories*, which represent a policy governing a group of QoS requirements specific to a particular environment such as time-critical communications; and
- *QoS management functions*, which can be combined in various ways and applied to various QoS characteristics in order to meet QoS requirements.

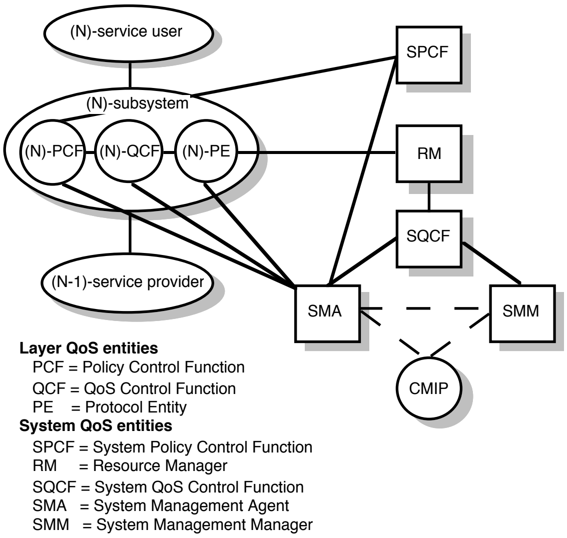
(N)-service user
(N)-subsystem
(N)-PCF
(N)-QCF
(N)-PE
(N-1)-service provider
SPCF
RM
SQCF
SMA
SMM
CMIP
Layer QoS entities
PCF = Policy Control Function
QCF = QoS Control Function
PE = Protocol Entity
System QoS entities
SPCF = System Policy Control Function
RM = Resource Manager
SQCF = System QoS Control Function
SMA = System Management Agent
SMM = System Management Manager

Figure 5-1. OSI QoS Framework

The OSI QoS framework (illustrated above) is made up of two types of management entities that attempt to meet the QoS requirements by monitoring, maintaining and controlling end-to-end QoS:

- i) *layer-specific entities*: The task of the policy control function is to determine the policy which applies at a specific layer of the open system. The policy control function models any priority actions that must be performed to control the operation of the layer. The definition of a particular policy is layer-specific and therefore cannot be generalized. Policy may, however, include aspects of security, time-critical communications and resource control. The role of the QoS control function is to determine, select and configure the appropriate protocol entities to meet layer-specific QoS goals.
- ii) *system-wide entities*: The system management agent is used in conjunction with OSI systems management protocols to enable system resources to be remotely managed. The local resource manager represents end-system control of resources. The system QoS control function combines two system-wide capabilities: to tune performance of protocol entities and to modify the capability of remote systems via OSI systems management. The OSI systems management interface is supported by the systems management manager which

provides a standard interface to monitor, control and manage end-systems. The system policy control function interacts with each layer-specific policy control function to provide an overall selection of QoS functions and facilities.

At Columbia University, Flossi and Yemini [82] have developed a *Quality Assurance Language (QuAL)* for the specification of QoS constraints on underlying computing and communication platforms. The specifications are compiled into run-time components that monitor the delivered QoS. Any QoS violations are fileded via user-level exception handlers. QuAL creates and manages a QoS-based MIB on a per- application basis for the management of flow statistics. The implementation work based in Concert-C and ST-II networking. The network level QoS specifications are detailed for both senders and receivers. A distributed application is viewed by QuAL as a set of autonomous processes that communicate by message exchange (Concert-C). At the application level, QuAL uses a contract identifier to present a set of constraints that a communication port must comply with. Only ports with compatible QoS attributes are connected.

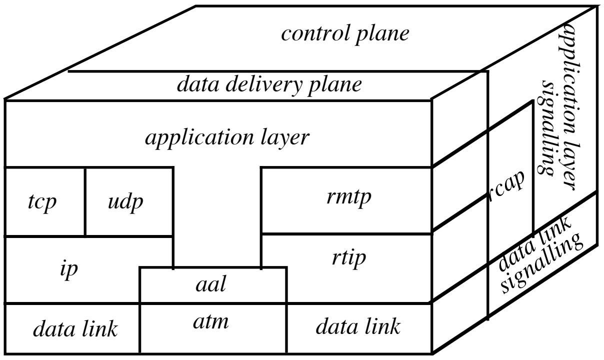

Figure 5-2. Tenet Architecture (inc. Internet Suite of Protocols)

The Tenet Group at the University of California at Berkeley have developed a family of protocols [83] [84] which run over an experimental wide area ATM network. The protocol family (as illustrated in Figure 5-2) includes a Real Time Channel Administration Protocol (RCAP) [61] in addition to Real Time Internet Protocol (RTIP), Continuous Media Transport Protocol (CMTP) [46]. The former provides generic connection establishment, resource reservation and signaling functions for the rest of the protocol family. RCAP spans the transport and network layers for overall resource reservation and flow setup. CMTP is explicitly designed for continuous media support. It is a lightweight protocol which runs on top of RTIP and provides sequenced and periodic delivery of continuous media samples with QoS control over throughput, delays and error bounds. The client interface to CMTP includes facilities to specify traffic characteristics in terms of burstiness, which is useful for variable bit rate encoding techniques, and workahead, which allows the protocol to deliver faster than the nominal rate if data is available. The Tenet Group [12] make a distinction between deterministic and statistical guranatees for hard real-time and continuous media flows respectively. In the deterministic case, guarantees provide a hard bound on the performance of all cells within a session. Statistical guarantees promise that no more than an x% of packets would experience a delay greater than specified, or no more that x% of cells might in a session might be lost.

The Tenet Architecture includes an application layer signalling protocol spans the end-system and the network, and provides QoS mapping between the application, transport and network layers; translating QoS constraints at each layer into a form which is needed by resource reservation protocols RCAP. The architecture also includes a scheme for dynamically managing real-time channels called DCM (Dynamic Connection Management) which supports media scaling (i.e., QoS adaptation). The motivation that underpins DCM is to increase network availability and flexibility. The adaptation can be initiated by the application or the network. For full details on modification contract and adaptation algorithms see [84]. Dynamic Connection Management guarantees either a transition from a primary to an alternative channel without any bound violations or a transition where a number of packets involved in a performance violation is bounded. Recently the Tenet Group suite of protocols are evolving to support multicast flows with heterogeneous QoS constraints [83].

The *HeiProject* at IBM's European Networking Center in Heidelberg have developed a comprehensive QoS model which provides guarantees in the end-systems and network [62]. The communications architecture includes a continuous media transport systems (HeiTS/TP) [47] which provides QoS mapping and media scaling [85]. Underlying the transport is an internetworking layer based on ST-II which supports both guaranteed and statistical levels of service; in addition the network supports QoS-based routing (via a QoS finder algorithm) and QoS filtering. Key in providing end-to-end guarantees is *HieRAT (resource administration technique)*: (based on initial work in [86]) a comprehensive QoS management scheme which includes QoS negotiation, QoS calculation, admission control and QoS enforcement, and resource scheduling [62]. The HeiRAT scheduling policy used in the supporting operating system is a rate-monotonic scheme whereby the priority of an operating system thread performing protocol processing is proportional to the message rate accepted.

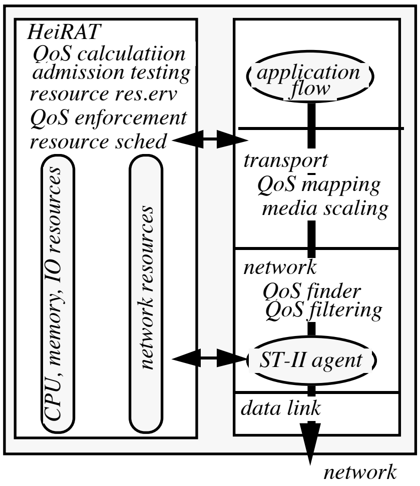

Figure 5-3. HeiProjects End-to-End QoS Model

Clark [87] introduces some early work on a *Quality of Service Manager (QM)* for an integrated services Internet suite of protocols. The QM (illustrated in Figure 5-4) presents an abstract management layer designed to isolate applications from underlying details of specific services provided in an QoS-driven Internet [57]. One motivating factor behind the introduction of a QM is applications can negotiate desired QoS without needing to know the details of a specific network services; in this case, the QM provides a degree of transparency whereby applications express desired levels of QoS in user-oriented language rather than using communication specifics. The QM is responsible for determining what QoS management capabilities are available on the application's communication path, and choosing the path best suited to the application. A number of benefits are gained by migrating specific services knowledge from to the application to the QM:

- *heterogeneity* is supported; the QM can match application needs to the underlying QoS capability;
- *transparency* is provided; applications will not need to be aware of the details of specific QoS management capability; and
- *extensibility* is supported; new QoS capabilities can be more easily deployed in the Internet, because applications will not be modified as new services become available.

The initial thrust of the work will be to map application specific needs to one of the new set integrated services (e.g., [102]) and provide some support for monitoring of performance. In the future, however, the QoS interface between the application and QM may cover more general issues such as cost of service, as well as more technical matters such as delay and bandwidth. In related work, Partridge [87] presents a multimedia-based Berkeley Sockets specification which includes support for a flows in terms of a flow-spec, and QoS management aspects of Clark's QM

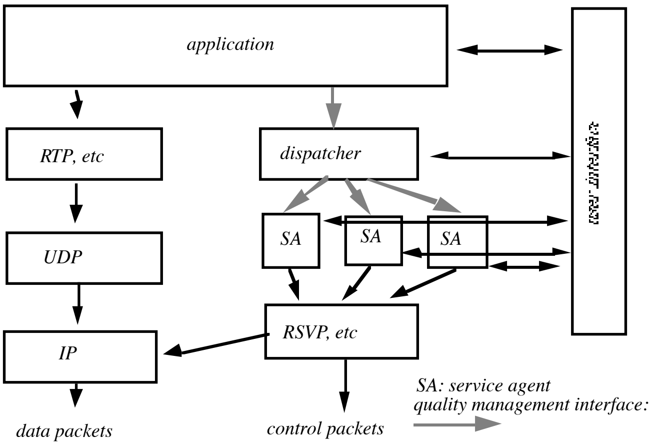

Figure 5-4. Integrated Services QoS Manager

Significant work carried out at the University of Pennsylvania describes a brokerage model [88] which incorporates QoS translation, and QoS negotiation and renegotiation (see [89] for full details on similar work on QoS negotiation protocol at University of Montreal). The notion of eras is introduced in [88] to describe variations in QoS parameters for complex, long-lived applications. Negotiation and renegotiation provide a mechanism to signal variations in QOS performance parameters at the user–network interface. They are invoked at era boundaries, and can aid resource allocation. In the model, application requirements and network resource allocation are expressed in fundamentally different terms and languages. A key part of the model, called a *QoS Broker* [88] is responsible for the translation of QoS at the user–network interface. More recent work as tackled operating system issues such as admission control for guaranteed QoS [90]; see also [89] for a comphrensive survey of resource management issues in netwroked multimedia.

Recently a number of QoS models have emerged from the distributing systems community. These include the Integrated *Multimedia Application Communication (IMAC) architecture* [92] and *ANSA QoS Framework* [74] which provides a model for real-time QoS. The IMAC architecture is based on the ANSA [32] architecture and has been implemented as an extension to the ANSA Testbed. IMAC provides a mechanism for specification of communication oriented QoS on per-invocation basis; interface operations may specify as set of QoS options. These QoS options are mapped to the underlying communications protocols as a set of QoS constraints on streams or bounded RPC communications. The work on the ANSA QoS Framework facilitates the enforcement of stringent time constraints found in distributed real-time applications.The model provides QoS specification and QoS-based binding for real-time programming in ANSA. The model, moreover, incorporates task and communication channels as its basic programming abstractions. It synthesizes aspects of resource requirements, resource allocation and resource scheduling into an object-based programming paradigm.

A number of projects are looking at providing enhanced services and mechanisms for open systems: *EuroBridge QoS-driven Architecture* [93] and *QoS-based Adaptive Architecture for packet scheduling (Q-ADAPTS)* [94]. Like the OSI QoS Framework, Q-ADAPTS and EuroBridge concentrate on quality of service for OSI communications. The EuroBridge research focuses on the upper layer architecture and the transport protocols, and provides unifying concepts for the management of QoS. The ADAPTS model developed at California State University and the Aerospace Corporation supports dynamic management of flows through scheduling algorithms and resource reservation. The work is based on the integration of existing OSI-based protocols and real-time scheduling mechanism in the end-system and network. The model addresses many limitations regarding QoS management which exist in the current OSI standards.

Projects carried out at Wollongong University, GMD FOKUS Berlin and Rutgers have designed new QoS models which take application level QoS requirements and map them down to ATM based networks. The *Simplified QoS Model* developed at GMD limits the number of performance parameters which the user can select. In doing so it simplifies the QoS mapping function and negotiation protocol. In [95] Damaskos and Gavras argue that applications can not be expected to configure large numbers of QoS parameters for flows. Judge and Beadle [96] from Wollongong University describe a QoS model for an ATM capable end-system connected to low speed ATM network (i.e., ranging 512 kbps to 2.4 kbps). The architecture supports two network traffic classes (guaranteed and best effort) for audio and data services. QoS mechanism for monitoring are built on AAL protocols. Degradation in the requested QoS can be forward to the application-level where compensatory actions can be taken. [97] reports on the development of the *GRAMS* architecture based on client/server paradym for QoS control of flows over ATM Networks. The major goal of a GRAMS is to serve heterogeneous QoS demands of clients exploiting the end-system and network utility. End-to-end resource management is based on a set of starvation counters used to measure system resource utilisation and individual flow QoS. These counters are integral to admission and rate control algorithms.

The Technical University of Berlin [98] is developing a QoS-based architecture with particular focus on support for the transport subsystem. The XTPX transport protocol, developed as part of the CIO (Coordination, Implementation and Operation of Multimedia Teleservices) Project, is at the heart of the architecture. The work considers QoS contracts, which are binding agreements between the application and transport provider, application classes, which flows are mapped into, and QoS management for the maintenance of flows. A multi-layer architecture is described which includes XTPX operating over IP and ATM. Cross layer functions are used to map QoS between layers and for resource management; multimedia synchronisation is implied in the literature.

Significant work at Washington University by Gopal and Purulkar [63] has developed a concrete solution to the problem of providing QoS guarantees in multimedia capable end-system architecture. The research considers QoS specification, QoS mapping and QoS enforcement (i.e., rate shaping) as fundamental end-system QoS mechanisms integrated into the protocol implementation model. The notion of QoS within the end-system is extended from the network interface driver, through the protocol layers and upto the application treads that generate/consume media.

Several projects in the European funded RACE program are concerned with QoS for integrated broadband networks. A significant contribution has been made by the QOSMIC (R.1082) project which studied QoS concepts in broadband networks, focusing on the user–network interface in particular. The major goal of the project was the specification of a QoS model for service life-cycle management which maps the user communication requirements to network performance parameters in a methodical manner. In some related work Jung and Seret [100] propose a framework for the translation of the performance parameters between the ATM Adaptation Layer (AAL) and ATM layers. They extend the QOSMIC model to include QoS verification. In this case the user can verify whether the achieved bearer QoS provided by the ATM network meets the contracted requirements expressed in terms of performance parameters. In related work the TOMQAT project [103] is developing the concept of total quality management in the context of broadband networks, analysing the end user quality of service requirements, and designing QoS control and management mechanisms to meet end-to-end QoS guarantees.

Media scaling [85] and codec translation [48] at the end systems, and filtering media traffic [99] [101] [62] and resource sharing [60] [84] in the network are very topical areas of QoS research at the moment. Media scaling matches the source with the receivers' QoS capability by manipulating flows at the network edges. In contrast, filtering accommodates the receivers' QoS capability by manipulating flows at the core of the network as they traverse bridges, switches and routers. Both schemes compensate for variation in network load/performance by re-scaling or filtering the delivered QoS respectively. Potentially this includes manipulating hierarchical flows; for example, delivering the I frames of an MPEG encoded flow and dropping the P and B frames to match the end system or network QoS constraint. Network level filtering looks very promising when used in conjunction with multicast protocols for dissemination of continuous media in support of heterogeneous receivers; for example, Pasquale et. el. [99] suggest that several receivers having disparate QoS communication requirements, and needing to access the same video flow simultaneous can be supported by a propagating filter scheme which deliver the appropriate QoS to each receiver. This scheme promotes efficient use of network resources, and as the literature suggests, reduces the likelihood of the on-set of congestion.

6. Comparison

In this section we present a summary of the distinct features of the QoS research addressed in this paper. For each QoS element of a the generalised QoS Framework we list the papers that cover the topic. The goal to provide a simple qualitative comparison.The legend for the comparison table is as follows: - indicates "not addressed"; E and N indicates "addressed in detail" in the end-system and network respectively; (E) and (N) indicates "mentioned only" in the end-system and network respectively; R indicated QoS renegotiation; S indicates scaling; D indicates QoS degradation; Sig indidcates QoS signal. The term "E2E coordination" refers to the coordination of end-system and network resources which would be implemented by a resource reservation, connection setup or signalling protocol.

QoS Model [Ref]	QoS Provision			QoS Control					QoS Management	
QoS Mechanism	QoS Mapping	Adm. Control / Resource allocation	E2E Coordi nation	Flow Scheduling	Flow Shaping	Flow Control	QoS Filtering	Flow Synchro nization	Monito ring / Alerts	QoS Mainte nance
XRM [67]	E N	E N	(E) N	(E) N	-	N	-	-	N	-
QoS-A [68]	E N	E (N)	E N	E (N)	E	(E)	(E) N	E	E Sig D	E N R S
TINA [71]	(E)	(N)	N	-	-	-	-	(N)	(N)	-
MASI [75]	E (N)	E (N)	E	E	-	-	-	E	E	E
TENET[83]	E N	N	N	N	N	(E)	N	-	E D	E R S
Broker [88]	E, (N)	E, (N)	E (N)	E	-	-	-	-	-	E R
QuAL [82]	E N	-	E (N)	-	-	-	-	-	E	E N R
WashU [63]	E	E		E	E	-	-	-	E	E R
UMont [89]	E N	E N	E N	-	-	-	-	-	-	E N R
isoQoS [80]	(E) (N)	E N	E N	-	-	-	-	-	E N	E N
Hei [62]	(E) (N)	E N	E N	E (N)	(E)	(N)	N	-	E D	E R S
Q-adapt[94]	(E) (N)	(E) (N)	E N	N	-	-	-	-	E	E N R
Grams [97]	-	E	-	E	E	-	-	-	E	E R
ieftQM [87]	E N	-	E	-	-	-	-	-	E N	E N R
IMAC [92]	E N	-	E N	-	-	-	-	E	E	E R
ANSA[74]	E N	E N	E N	-	-	-	-	E	E	E R
EuroB [93]	(E) (N)	-	(E) (N)	-	-	-	-	-	E N	E N
Simple [95]	E N	E N	E N	-	E N	(N)	-	-	E	E R
NEC [4]	(E) (N)	(E) (N)	(E) (N)	-	-	(E)	-	(E) (D)	(E) (R)	(E) (R)
WollU [96]	(E)	-	(E) (N)	-	-	-	-	-	E	(E) (R)
XTPX[98]	E, N	-	E (N)	-	-	E	-	E	E	-

Table 1: Comparison of QoS Models

7. Conclusion

In this paper we have argued that systems designers should adopt an end-to-end approach to meet application level QoS requirements. To meet this challenging goal, all components of distributed systems architecture must work together in unison. While the area of QoS research in multimedia networking is mature, work on QoS architecture remains in its early stage of development. The work presented in this paper contributes towards an understanding of the key principles, services and mechanisms needed to build quality of service into networked multimedia systems.

8. References

[1] Hutchison, D., Coulson G., Campbell, A., and G. Blair , “Quality of Service Management in Distributed Systems”, M. Sloman ed., Network and Distributed Systems Management, Addison Wesley, chapter 11, 1994.

[2] Campbell, A., Coulson, G., García, F., Hutchison, D., and H. Leopold, “Integrated Quality of Service for Multimedia Communications”, Proc. IEEE INFOCOM’93, pp. 732-739, San Francisco, USA, April 1993.

[3] Lazar, A.A., “A Real-time Control, Management, and Information Transport Architecture for Broadband Networks,” Proc. International Zurich Seminar on Digital Communications, pp. 281-295, 1992.

[4] Bansal, V., Siracusa, R.J, Hearn, J. P., Ramamurthy and D. Raychaudhuri, “Adaptive QoS-based API for Networking, Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire, April, 1995.

[5] Saltzer, J., Reed, D., and D. Clark, "End-to-end Arguments in Systems Design", ACM Trans. on Computer Systems, Vol. 2., No. 4., 1984.

[6] Tennenhouse, D.L., "Layered Multiplexing Considered Harmful", Protocols for High-Speed Networks, Elsevier Science Publishers (North-Holland), 1990.

[7] Clark, D., and D.L. Tennenhouse, "Architectural Consideration for a New Generation of Protocols", Proc. ACM SIGCOMM ‘90, Philadelphia, 1984.

[8] Chesson, G., “XTP/PE Overview”, Proc. 13th Conference on Local Computer Networks, Pladisson Plaza Hotel, Minneapolis, Minnesota, 1988.

[9] Zitterbart, M., Stiller, B., and A Tantawy,“A Model for Flexible High-Performance Communication Subsystems”, IEEE JSAC, May 1992.

[10] Little, T.D.C, and A. Ghafoor, “Synchronisation Properties and Storage Models for Multimedia Objects”, IEEE Journal on Selected Areas on Communications, Vol. 8, No. 3, pp. 229-238, April 1990.

[11] Partridge, C., "A Proposed Flow Specification; RFC-1363" Internet Request for Comments, no. 1363, Network Information Center, SRI International, Menlo Park, CA, September 1990.

[12] Ferrari D. and Verma D. C., “A scheme for real-time channel establishment in wide-area networks,” IEEE JSAC, 8(3), 368–77, 1990.

[13] Clark, D.D., Shenker S., and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism" Proc. ACM SIGCOMM’92, pp. 14-26, Baltimore, USA, August, 1992.

[14] Campbell, A., Coulson G. and D. Hutchison, “Supporting Adaptive Flows in a Quality of Service Architecture, ” Multimedia Systems Journal, November, 1995.

[15] Kelly, F.P., "On Tariffs, Policing and Admission Control for Multiservice Networks", Proc. Multiservice Networks ‘93, Cosener’s House, Abingdon, July 1993, and Internal Report, Statistical Laboratory, University of Cambridge, England, 1993.

[16] Coulson, G., Campbell, A and P. Robin, “Design of a QoS Controlled ATM Based Communication System in Chorus,” IEEE Journal of Selected Areas in Communications (JSAC), Special Issue on ATM LANs: Implementation and Experiences with Emerging Technology, May 1995.

[17] Parekh, A. and R. G. Gallager, "A Generalised Processor Sharing Approach to Flow Control in Integrated Service Networks

- The Multiple Node Case", Proc. IEEE INFOCOM'93, pp.521-530, San Francisco, USA, April 1993.

[18] Keshav, S., “On the Efficient Implementation of Fair Queueing”, Internetworking: Research and Experiences, Vol. 2, pp 157-173, 1991.

[19] C. Liu, J. Layland, “Scheduling Algorithms for Multiprogramming in Hard Real Time Environment”, Journal of the ACM, 1973.

[20] Stankovic et al., “Implications of classical scheduling Results for Real-Time Systems,” IEEE Computer, Special Isssue on Scheduling and Real-Time Systsems, June 1995.

[21] Tokuda H. and T. Kitayama ,”Dynamic QOS Control Based on Real-Time Treads” Proc. Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster University, Lancaster LA1 4YR, UK, 1993.

[22] H. Zhang, S. Keshav, “Comparison of Rate-Based Service Disciplines”, ACM SIGCOMM, 1991.

[23] ATM Forum, ATM User-Network Inteface Specifications, Version 3.0, Prentice-Hall, 1993.

[24] Shenker, S., Clark, D., and L. Zhang, (1993) "A Scheduling Service Model and a Scheduling Architecture for an Integrated Service Packet Network", Working Draft available via anonymous ftp from parcftp.xerox.com: /transient/service-model.ps.Z.

[25] Jacobson, V., (1994) "VAT: Visual Audio Tool", vat manual pages, 1993.

[26] Kanakia, H., Mishra, P., and A. Reibman, (1993) "An Adaptive Congestion Control Scheme for Real Time Packet Video Transport", Proc. ACM SIGCOMM ‘93, San Francisco, USA, October 1993.

[27] Escobar, J., Deutsch, D. and C. Partridge, “Flow Synchronisation Protocol,” IEEE GLOBECOM'92, Orlando, Fl., December 1992.

[28] Pacaifici, G., and R. Stadler, “An Architecture for Performance Management of Multimedia Networks,” Proc. IFIP/IEEE International Symposium on Integrates Network Management, Santa Barbara, May 1995.

[29] Vogel, A., G. v. Bochmann, R. Dssouli, J. Gecsei and B. Kerherv, “Distributed Multimedia Applications and Quality of Service - A Survey,” IEEE Multimedia, 1994.

[30] Miloucheva, I. “Quality of Service Research for Distributed Multimedia Applications,” ACM Pacific Workshop on Distributed Multimedia Systems, 1995.

[31] Mullender S., ed. (1993). Distributed Systems, 2nd edn., Addison-Wesley.

[32] APM Ltd , “ANSAware 3.0 Implementation Manual,” APM Ltd, Poseidon House, Castle Park, Cambridge CB3 0RD, UK, 1991

[33] Open Software Foundation,”Distributed Computing Environment,” 11 Cambridge Center, Cambridge, MA 02142, USA, 1992.

[34] ODP, “Draft recommendations X.903: basic reference model of open distributed processing”, ISO/IEC JTC1/SC21/WG7, International Standards Organisation, 1992.

[35] Coulson G., Blair G. S., Davies N. and Williams N.” Extensions to ANSA for multimedia computing,” Computer Networks and ISDN Systems, 25(11), 305–23, 1992.

[36] Nicolaou, C., “An Architecture for Real-Time Multimedia Communication Systems”, IEEE Journal on Selected Areas in Communications, Vol. 8, No ¡3, April 1990.

[37] Hazard, L., Horn, F., and J. B. Stefabi, “Towards the Integration of Real-Time and QoS Handling in ANSA”, CNET Report CNET.RC.ARCADE.01, June 1993.

[38] Zinky, J., Bakken, D., R. Schantz, “Overview of Quality of Service for Distributed Objects” Technical Report, BBN Systems and Technologies, Cambridge, 1995.

[39] Hayter, M. and D. McAurely, “The Desk Area Network”, ACM Operating Systems Review, Oct 1991.

[40] Feldmeier, D.,“A Framework of Architectural Concepts for High Speed Communication Systems”, Computer Communication Research Group, Bellcore, Morristown, May 1993.

[41] Hanko J. G., Keurner E. M., Northcutt J. D. and Wall G. A.," Workstation support for time critical applications," Proc. Second International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Springer Verlag, 1991.

[42] Bulterman D. C. and van Liere R.," Multimedia synchronisation and UNIX," Proc. Second International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Springer Verlag, 1991.

[43] Leslie, I.M., McAuely, D., and S.J. Mullender, "Pegasus - Operating Systems Support for Distributed Multimedia Systems," Operating Systems Review, Vol. 27, No. 1, 1993.

[44] Clark, D.D., Lambert, M.L., and L. Zhang, "NETBLT: A High Throughput Transport Protocol," Computer Communications Review, Vol. 17, No. 5, 1987.

[45] Danthine, A., Baguette Y., Leduc G., and L. Leonard, "The OSI 95 Connection-Mode Transport Service - Enhanced QoS", Proc. 4th IFIP Conference on High Performance Networking, University of Liege, Liege, Belgium, December 1992.

[46] Wolfinger, B. and M. Moran, "A Continuous Media Data Transport Service and Protocol for Real-time Communication in High Speed Networks." Second International Workshop on Network and Operating System Support for Digital Audio and Video, IBM ENC, Heidelberg, Germany, 1991.

[47] Hehmann, D.B., Herrtwich R.G., Schulz W., Schuett, T., 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.

[48] Schulzrinne, H. and S. Casner, "RTP: A Transport Protocol for Real-Time Applications", Work in Progress, Internet Draft, <draft-ietf-avt-rtp-05.ps>, 1995.

[49] Campbell A., Coulson G., Garcia F. and Hutchison D., "A Continuous Media Transport and Orchestration Service," Proc. ACM SIGCOMM '92, Baltimore, Maryland, USA, 99–110, 1992.

[50] W. Doeringer, D. Dykeman, M. Kaiserswerth, B. Meister, H. Rudin, R. Williamson, "A Survey of Light-weight Transport Protocols for High-speed Networks", IEEE Transactiond on Communications, November 1990.

[51] Keshav, S., "Report on the Workshop on Quality of Service Issues in High Speed Networks", ACM Computer Communications Review, Vol 22, No 1, pp 6-15, January, 1993.

[52] Kurose, J.F., "Open Issues and Challenges in Providing Quality of Service Guarantees in High Speed Networks", ACM Computer Communications Review, Vol 23, No 1, pp 6-15, January 1993

[53] Golestani, S.J., "A Stop and Go Queueing Framework for Congestion Management,"| Proc. ACM SIGCOMM'90, San Francisco, June 1990.

[54] Guerun, R., Ahmadi, H., and M. Naghshineh,"Equivalent Capacity and its Application to Bandwidth Allocation in High Speed Networks," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, Sept. 1991.

[55] Cruz, R., "A Calculus for Network Delay: Part I: Network Elements in Isolation," IEEE Transactions on Info. Theory, Vol. 37. No. 1, Jan. 1991.

[56] Hyman, J., Lazar, A., and G. Pacifici, "Real-Time Scheduling with Quality of Service Constraints", IEEE Journal on Selected Areas in Communications, Vol. 9. No. 7, April 1990.

[57] Braden R., Clark, D., and S. Shenker,"Integrated Services in the Internet Architecture: an Overview", Request for Comments, RFC-1633, 1994.

[58] Floyd, S., "Link-Sharing and Resource Management Models for Packet Networks", Draft available via anonymous ftp from ftp.ee.lbl.gov: link.ps.Z, September 1993.

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

[60] Zhang, L., et. al., "RSVP Functional Specification", Working Draft, draft-ietf-rsvp-spec-03.ps, 1995.

[61] A. Benerjea and B. Mah, "The Real-Time Channel Administration Protocol", 2nd International Workshop on Network and Operating System Support for Digital Audio and Video", Heidelberg, November 1991.

[62] Volg, C., Wolf, L., Herrtwich, R. and H. Wittig, “HeiRAT - Quality of Service Management for Distributed Multimedia Systems”, Multimedia Systems Journal, November 1995.

[63] Gopalakrishna, G., and G. Parulkar, “Efficient Quality of Service in Multimedia Computer Operating Systems”, Department of computer science, Washington University, Report WUCS-TM-94-04, August 1994.

[64] Lazar, A. A., “Challenges in Multimedia Networking”, Proc. International Hi-Tech Forum, Osaka, Japan, Februray 1994.

[65] Hyman, J., Lazar, A., and G. Pacifici, "Joint Scheduling and Admission Control for ATS-based Switching Nodes", Proc. ACM SIGCOMM ‘92, Baltimore, Maryland, USA, August 1992.

[66] Lazar, A. A., Ngoh, L.H. and A. Sahai, “Multimedia Networking Abstraction with Quality of Services Guarantees,”l Proc. SPIE Conference on Multimedia Computing and Networking, San Jose, February 1995.

[67] Lazar, A. A., Bhonsle S., Lim, K.S., "A Binding Architecture for Multimedia Networks", Proceedings of COST-237 Conference on Multimedia Transport and Teleservices, Vienna, Austria, 1994.

[68] Campbell, A., Coulson, G. and D. Hutchison, “A Quality of Service Architecture,” ACM Computer Communications Review, April 1994.

[69] Nilison, G., Dupuy, F., and Chapman, “An Overview of the Telecommunications Information Networking Architecture,” Proc. TINA’95, Melbourne, 1995.

[70] OMG, (1993), “The Common Object Request Broker: Architecture & Specification, Rev 1.3., December 1993.

[71] TINA-C, “The QoS Framework”, Internal Technical Report,1994.

[72] Leydekkers, V. Gay and L. Franken, “A computational and engineering view on Open Distributed Real-time Multimedia exchange,” Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire, 1995.

[73] TINA-C, “The DPE Kernel”, Internal Technical Report,1995.

[74] Guangxing, “An Model of Real-Time QoS for ANSA,” Technical Report APM.1151.00.04, APM Ltd, Cambrigde, UK, March 1994.

[75] Besse, L., Dairaine L., Fedaoui, L., Tawbi, W., and K. Thai, “Towards an Architecture for Distributed Multimedia Application Support”, Proc. International Conference on Multimedia Computing and Systems, Boston, May 1994.

[76] L. Fedaoui, A. Seneviratne and E. Horlait, “Implementation of a End-to-End Quality of Service Management Scheme”, Cost 237 Workshop, Vienne, November 1994.

[77] M. Fry, A. Seneviratne, A Richards, “Framework for the Implementation of the the Next Generation of Communication Protocols”, In 4th International Workshop on Network and Operating Systems Support for Digital Audio and Video, University of Lancaster, November 1993.

[78] Santoso, H., Dairaine, L., Fdida, S., and E. Horlait, “Preserving Temporal Signature: a Way to Convey Time Constrained Flows”, IEEE Globecom, November 1993.

[79] J. P Lehoczky, L. Sha, Y. Ding, “The Rate Monotonic Scheduling Algorithm: Exact Characterisation and Average Case Behaviour”, 10th IEEE Real-Time Symposium, 1989.

[80] ISO-QoS, "Quality of Service Basic Framework - Qutline", ISO/IEC JTC1/SC21/WG1 N1145, International Standards Organisation, UK, 1994.

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

[82] Flossi, P. G. S., and Y. Yemini,” QuAL: Quality Assurance Language,” ITS’94.

[83] Ferrari, D., Ramaekers J. , and G. Ventre, "Client-Network Interactions in Quality of Service Communication Environments”, Proc. 4th IFIP Conference on High Performance Networking, University of Liege, Liege, Belgium, December 1992.

[84] Ferrari, D., “The Tenet Experience and he Design of Protocols for Integrated Services Internetworks,” Multimedia Systems Journal, November 1995.

[85] Delgrossi, L., Halstrinck, C., Hehmann, D.B., Herrtwich R.G., Krone, J., Sandvoss, C., and C. Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System", Proc. ACM Multimedia ‘93, Anaheim, August 1993.

[86] Anderson, D.P., Herrtwich R.G., and C. Schaefer. "SRP: A Resource Reservation Protocol for Guaranteed Performance Communication in the Internet", Internal Report , University of California at Berkeley, 1991.

[87] Int-svr slides, ftp//

[88] Nahrstedt K. and J. Smith, “The QoS Broker”, IEEE Multimedia, Spring 1995.

[89] Vogel, A., G. v. Bochmann, R. Dssouli, J. Gecsei, A. Hafid and B. Kerherve, “On QoS Negotiation in Distributed Multimedia Application”, Proc. Protocol for High Speed Networks, April 1994.

[90] Nahrstedt K. and J. Smith, “A Service Kernel for Multimedia Endstations”, Proc. IWACA’94: Multimedia: Adavnced Teleservices and High-Speed Communication Architectures, Heidelberg 1994.

[91] Nahrstedt, K., and R. Steinmetz, “Resource Management in Networked Multimedia Systems”, K. Nahrstedt and R. Steinmetz, IEEE Computer Magazine, May 1995.

[92] Nicolaou, C., “Integrating Multimedia into the ANSA Architecture, “ Tecnical Report TR.028.93, APM Ltd, Cambridge, UK. 1993.

[93] Pronios, N., “EuroBridge: A QoS-Driven Architecture,” Technical Report, Intracom S.A, Greece, 1995.

[94] Tran, V., and T. Bradley Maples, “An Adaptive Model for Real-Time Management of Quality of Service in the OSI Reference Model,” ICC’95, Seattle, 1995.

[95] Damaskos, S. and A. Gavras, “A Simplified QoS Model for Multimedia Protocols over ATM”, High Peformance Networking, S.Fdida ed., Elsevier Scince B. V. (North-Holland), 1994.

[96] Judge, J., and P. Beadle, “Supporting Quality of Service on Multimedia Terminals Interconnected by a Low Speed ATM Network”, SPIE Vol. 2417, 1995.

[97] Hui, J., Zhang, J., and Jun Li, “Quality of Service in GRAMS for ATM Local Area Networks”, IEEE Journal of Selected Areas in Communications (JSAC), Special Issue on ATM LANs: Implementation and Experiences with Emerging Technology, May 1995.

[98] Miloucheva, I.and K. Rebensburg, “QoS-based Architecture using XTP”, 4th IEEE International Conference on Future Trends of Distributed Systems, Lisboa, Sept, 1993.

[99] Pasquale G., Polyzos E., Anderson E. and Kompella V. The multimedia multicast channel. Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, USA, 1992.

[100] Jung, J., and D. Seret , “Translation of QoS Parameters into ATM Performance Parameters in B-ISDN”, Proc. IEEE Infocom’93, Vol. 3, San Francisco, USA, 1993.

[101] Yeadon, N., Garcia, F., Campbell, A and D. Hutchison, “QoS Adaptation and Flow Filtering in ATM Networks”, 2nd International Workshop on Advanced Teleservices and High Speed Communication Architectures, Heidelberg, 1994.

[102] Shenker, S., and C. Partridge (1995), “Specification of Predictive Quality of Service”, Working Draft, draft-ietf-intserv-predictive-svc-00.txt.

[103] TOMQAT, Deliverables, ftp://ftp.fokus.gmd.de/pub/race/tomqat

[104] M. Fry, A. Seneviratne, A Richards, “Framework for the Implementation of the the Next Generation of Communication Protocols”, In 4th International Workshop on Network and Operating Systems Support for Digital Audio and Video, University of Lancaster, November 1993.