Multimedia Systems (1998) 6: 167–178

Multimedia Systems

Transporting QoS adaptive flows

Andrew T. Campbell[1], **Geoff Coulson**[2], **David Hutchison**[2]

[1] Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, New York, NY 10027-6699, USA; http://www.ctr.columbia.edu/~campbell
[2] Department of Computing, Lancaster University, Lancaster LA1 4YR, UK; e-mail:{geoff,dh}@comp.lancs.ac.uk

Abstract. Distributed audio and video applications need to adapt to fluctuations in delivered quality of service (QoS). 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 made to adapt to fluctuating QoS with minimal perceptual distortion. In this paper, we extend our previous work on a QoS Architecture (QoS-A) by populating the QoS management planes of our architecture with a framework for the control and management of multilayer coded flows operating in heterogeneous multimedia networking environments. Two key techniques are proposed: i) an end-to-end rate-shaping scheme which adapts the rate of MPEG-coded flows to the available network resources while minimizing the distortion observed at the receiver; and ii) an adaptive network service, which offers "hard" guarantees to the base layer of multilayer coded flows and "fairness" guarantees to the enhancement layers based on a bandwidth allocation technique called Weighted Fair Sharing.

Key words: Scalable flows – Multimedia transport – Dynamic QoS management – end-to-end QoS architecture

1 Introduction

The interplay between user-oriented quality-of-service (QoS) requirements [Zhang,95], multilayer coded flows [Shacham, 92], and end-to-end communication support [Campbell,95] is an interesting and active area of research [Pasquale,93; Delgrossi,93; Tokuda,92; Hoffman,93]. The hierarchical coding techniques used by coders such as MPEG-1, MPEG-2 and H261 make possible a range of creative adaptation strategies, whereby fluctuating bandwidth availability can be accommodated by selectively adding/ removing coding layers.

In this paper, we introduce the concept of *dynamic QoS management* (DQM) which, by exploiting and extending the above principles, controls and manages multilayer coded flows operating in heterogeneous, multicast, multimedia networking environments. Under the DQM heading, two main techniques are proposed to support scalable[1] video over multimedia networks. These are i) *end-to-end rate shaping*, which adapts the rate of multilayer flows to the available network resources while minimizing the distortion observed at the receiver, and ii) an *adaptive network service*, which offers a combination of "hard" guarantees to the base layer of multilayer coded flows and "fairness" guarantees to the enhancement layers based on a new bandwidth allocation technique called *weighted fair sharing*. We also discuss a number of types of distributed scaling object, which are used to manage and control QoS. These include *QoS filters*, which manipulate hierarchically coded flows as they progress through the communications system, *QoS adaptors*, which scale flows at end-systems based on the flow's measured performance and user-supplied QoS scaling policy, and *QoS groups*, which provide baseline QoS for multicast flows.

All these components are inter-related in a QoS Architecture (QoS-A) [Campbell,94], which has been developed over the last 4 years [Campbell,93]. This research has been conducted in co- operation with ATM switch manufacturer GDC (formally Netcomm Ltd). The QoS-A is a layered architecture of services and mechanisms for QoS management and control of continuous media flows in multiservice networks. The architecture incorporates the following key notions: *flows* characterize 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 realization of the flow concept demands active QoS management and tight integration between device management, thread scheduling, communications protocols and networks.

The structure of the paper is as follows. We first present, in Sect. 2, background on the QoS-A. Following this, we describe the salient features of scalable video flows in Sect. 3 before describing, in Sect. 4, the set of scaling objects used in our architecture together with the application program-

Correspondence to: A.T. Campbell

[1] *Media scaling* is a general term, first proposed by [Delgrossi,93], we use to refer to the dynamic manipulation of media flows as they pass through a communications channel.

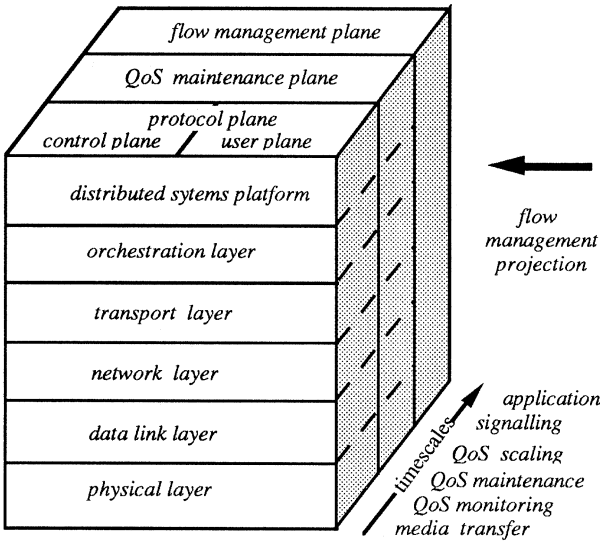


Fig. 1. QoS-A

ming interface (API) to DQM. Section 5 then presents our scheme for end-to-end DQM. In this section, the detailed operation of a number of specific types of scaling object is described. Following this, Sect. 6 introduces our adaptive network service, defines the notion of weighted fair-share resource allocation, explains the rate control scheme used in the network, and describes the use of some network-oriented QoS filters. Finally, in Sect. 7, we present the status of our work and offer some concluding remarks.

2 Quality-of-service architecture (QoS-A)

The QoS-A is based on a set of principles that govern the realization of end-to-end QoS in a distributed systems environment:

i) the *transparency principle* states that applications should be shielded from the complexity of handling QoS management [Campbell,92];
ii) the *integration principle* states that QoS must be configurable, predicable and maintainable over all architectural layers to meet end-to-end QoS [Campbell,93];
iii) the *separation principle* states that information transfer, control and management are functionally distinct activities in the architecture and operate on different time scales [Lazar,90]; and
iv) the *performance principle* guides the division of functionality between architectural modules (viz. end-to-end argument [Saltzer,84], application layer framing and integrated layer processing [Clark,90], and the reduction of layered multiplexing [Tennenhouse,90]).

In functional terms, the QoS-A (see Fig. 1) 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 a distributed object-based environment [Coulson,93]. Below the platform level is an *orchestration layer*, which provides jitter correction and multimedia synchronization 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. See [Campbell,94] for full details on the QoS-A.

QoS management is realized 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. We use separate protocol profiles for the control and data components of flows because of the different QoS requirements of control and data: control generally requires a low-latency full-duplex assured service, whereas media flows generally require a range of non-assured, high-throughput and low-latency simplex services. 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 synchronization 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).

Recent work on the QoS-A has concentrated on realizing the architecture in an environment comprising an enhanced Chorus micro-kernel [Coulson,95] and an enhanced multimedia transport service and protocol [Campbell,94] in the local ATM environment.

3 Scalable video flows

The fundamental design goal of digital audio-visual information representation schemes in the past several decades has been that of *compactness*: describe the signal's content with as few bits as possible. The algorithmic foundation of this work has been based on the assumption that information transport occurs over constant bandwidth and constant delay channels. This is an assumption that does not necessarily hold valid in an environment of packet switched networks working on the premise of *multiplexing gain*.

Our primary focus in this paper is MPEG-2 coded video. Within this framework, there are two alternative techniques which underpin QoS adaptation: *intrinsic* and *extrinsic* techniques. The former are provided by the encoder and are embedded in the coded bitstream. The latter are provided by QoS filters that operate directly on the compressed bitstream, performing the desired manipulations. Their difference lies in their complexity and performance. Intrinsic techniques can have a very simple implementation, but, as we will see, offer only a discrete range of possibilities. Extrinsic techniques are computationally more complex, but can operate

on a continuum of possibilities. In the following section, we briefly describe the architecture of MPEG-2, with particular emphasis on the intrinsic adaptation capabilities it provides in the form of *scalability profiles*. Extrinsic adaptation can be provided through the use of dynamic rate-shaping QoS filters [Eleftheriadis,95a] which are discussed in Sect. 5.2.1.

3.1 MPEG-2

The algorithmic foundation of MPEG is motion-compensated, block-based transform coding MPEG-2 [H.262,94]. Each block is transformed using the Discrete Cosine Transform (DCT), and is subsequently quantized. Quantization is the sole source of quality loss in MPEG-2 and is, of course, a major source of compression efficiency. The quantized coefficients are converted to a one-dimensional string using a zig-zag pattern and then run-length encoded. There are three types of pictures in a sequence: I, P, and B. I (or *intra-*) pictures are individually coded, and are fully self-contained. P pictures are *predicted* from the previous I or P picture, while B (or *bi-directional*) pictures are interpolated from the closest past and future I or P pictures. Prediction is motion-compensated; the encoder finds the best match of each macroblock in the past or future picture within a pre-specified range. The displacement(s), or motion vector(s), is sent as side information to the decoder.

In order to increase the coding efficiency, MPEG-2 relies heavily on entropy coding. Huffman codes (variable length codewords) are used to represent the various bitstream quantities (run-length codes, motion vectors, etc.). As a result, the output of an MPEG-2 encoder is inherently a *variable-rate* bitstream: the ratio of bits per pixel varies from one block to the next. In order to construct a constant-rate bitstream, *rate control* is used. This is achieved by connecting a buffer to the output of the encoder. The buffer is emptied at a constant rate and its occupancy is fed back to the encoder. This information is used to control the selection of the quantizer for the current macroblock. High buffer occupancy leads to more coarsely quantized coefficients, and hence less bits per block, and vice versa. Through this self-regulation technique one can achieve a constant output rate.

For transport purposes, video and audio are multiplexed according to the *system layer* of MPEG-2, which defines the packetization structure and synchronization algorithms between the audio and video signals. Two different packetization structures are defined, namely *program streams* and *transport streams*. Both are logically constructed from "packetized elementary stream" (PES) packets. These are the basic units in which individual audio, video, and control information are carried. Program streams are designed for use in relatively error-free environments, use variable length packets and combine PES packets that have a common time base. Transport streams are designed for noisy channels, utilize fixed-length packets (188 bytes) and can carry programs with independent time bases. The system layer's timing model is based on the assumption that a constant-delay transport mechanism is used. Although deviations from this are allowed, the way to address them is not specified. All timing information is based on a common system clock and timestamps (i.e., an absolute timing method) ensure proper inter-media synchronization between the audio and video signals upon presentation.

3.2 MPEG-2 scalability modes

From the above discussion it is clear that MPEG-2 is especially tuned for transmission over traditional constant bandwidth and delay channels. Some support for flexible and/or robust transmission is provided through the use of *scalability modes* for channels exhibiting dynamic behavior. The situation is even more challenging for scalable flows, i.e., in cases where the available bandwidth may vary over time. However, here the benefits of multiplexing gain are possible.

MPEG-2 provides for the simultaneous representation of a video signal at various different levels of quality through the use of multiple independent bitstreams or sub-signals. This is achieved through the use of pyramidal, or hierarchical coding – one first constructs a coarse or base representation of the signal, and then produces successive enhancements. The latter assumes that the base representation is available and only encodes the incremental changes that have to be performed to improve the quality. There are four different scalability modes: *spatial*, *SNR*, *temporal*, and *data partitioning*. MPEG-2 allows the simultaneous use of up to two different scalability modes (except from data partitioning) in any combination, hence resulting in a three-level representation of the signal. In spatial scalability, the base and enhancement layers operate at different spatial resolutions (e.g., standard TV and HDTV). In SNR scalability, both layers have the same resolution and the enhancement refines the quantization process performed in the base layer. In temporal scalability, the enhancement layer increases the number of frames per second of the base layer (e.g., from 30 frames per second to 60 fps). Data partitioning is slightly different from the other three scalability modes in the sense that the encoder does not maintain two different prediction loops, and hence the base layer is not entirely self-contained. Its benefit is that it can be applied even in single-layer encoders.

As an example, the spatial scalability mode can be used to transmit digital TV in both standard and HDTV formats. Although two sub-signals are actually generated, this is not identical to simulcast: the total bandwidth required is much smaller, since the HDTV layer uses the base, standard TV layer as a reference point. Scalability can be very useful in transmission of video over adaptive channels.

3.3 Discrete and continuous QoS adaptation

Although MPEG-2's scalability features are useful in resolving the heterogeneity problems described above [Delgrossi,93] and are useful in numerous applications, their use in continually QoS-varying channels is problematic. This is because they only allow the representation of the signal at a fixed number of discrete quality points (temporal or spatial resolution, or spatial quality). These points are typically significantly apart and transitions between the two are perceptually significant. Table 1 [Paek,95] shows an example of hybrid scalability with spatial (E1) and SNR (E2) enhancement layers.

Table 1. MPEG-2 hybrid scalable bitstream using spatial and SNR scalability (24 fps)

Layer name	Profile	Symbol	Frame size	Bit rate	Subjective QoS
Base layer	Main	BL	304×112	0.32 Mbps	VHS
Enhancement 1 layer	Spatial	E1	608×224	0.83 Mbps	Super VHS
Enhancement 2 layer	SNR	E2	608×224	1.85 Mbps	Laser disc

Consider, for example, a channel that temporarily undergoes rate variability for a period of a few seconds. Switching to a lower quality point by discarding the enhancement layer(s) for such a brief interval will create a perceptual "flash" that is annoying to users. An additional issue is that as soon as compression parameters are established it is impossible to modify them later after compression is completed. Hence, scalability modes can only really be used for well-defined, simple channels that vary slowly. Since the wide range of different access mechanisms to multimedia information makes it difficult to select *a priori* a set of universally inter-operable coding parameters, it is necessary to provide extrinsic mechanisms that allow the representation of the "signal at a continuum of qualities and rates" [Delgrossi,93] so that scalable flows can be accommodated.

This is possible through the use of a class of dynamic rate-shaping QoS filters [Eleftheriadis,95b] and the provision of adaptive network services providing a QoS continuum for fully scalable flows. The adaptive service introduced in this paper uses explicit feedback from network resource management to dynamically shape the video source based on available network resources. Some benefits of an adaptive scheme are non-reliance on video modelling techniques and statistical QoS specification and specific support for the semantics of scalable video flows (e.g., MPEG-2 scalable profiles). Dynamic rate-shaping filters manipulate the rate of MPEG-coded video, matching it to the available bandwidth (indicated by the adaptive service), while minimizing the distortion observed by the receiver.

4 Distributed scaling objects and API extensions

In this section, we introduce a set of *distributed scaling objects* used to manipulate hierarchically coded flows as they progress through the communications system. These comprise QoS adaptors, QoS filters and QoS groups as defined in Sect. 1. We also introduce an extension to the QoS-A application programmer's interface (API) which gives access to the scaling object types.

4.1 Scaling objects

4.1.1 QoS adaptors

QoS adaptors are used in conjunction with the QoS-A flow-monitoring function to ensure that the user QoS specified in the service contract is maintained. In this role, QoS adaptors are seen as QoS arbiters between the user and network. QoS adaptors scale flows at the end-systems based on a user-supplied QoS scaling policy (see Sect. 4.2) and the measured performance of on-going flows. QoS adaptation is discussed in detail in Sect. 5.

4.1.2 QoS filters

QoS filters manipulate multilayer coded flows [Shacham,92; Hoffman,93; Zhang,95] at the end-systems and as they progress through the network. We describe three distinct styles of QoS filters:

i) *shaping filters*, which manipulate coded video and audio by exploiting the structural composition of flows to match network, end-system or application QoS capability. Shaping filters are generally situated at the edge of the network at the source. They require non-trivial computational power. Examples are the dynamic rate-shaping (DRS) filter and the source bit rate (SBR) filter (see Sect. 5.2);
ii) *selection filters*, which are used for sub-signal selection and media dropping (e.g., video frame dropping) are of low complexity and low computational intensity, selection filters are designed to operate in the network and are located at switches. They require only minimal computational power. Examples are sub-signal filter, hierarch filter, hybrid filter (see Sect. 6.3); and
iii) *temporal filters*, which manipulate the timing characteristics of media to meet delay bound QoS are also low in complexity and trivial computationally. Temporal filters are generally placed at receivers or sinks of continuous media where jitter compensation or orchestration of multiple related media is required. Examples are sync filter, orch filter (see Sect. 5.3).

4.1.3 QoS groups

Before potential senders and receivers can communicate they must first join a *QoS group* [Aurrecoechea,95]. The concept of a QoS group is used to associate a baseline QoS capability to a particular flow. All sub-signals of a multilayer stream can be mapped into a single flow and multicast to multiple receivers [Shacham,92]. Then, each receiver can select to take either the complete signal advertized by the QoS group or a partial signal based on resource availability. Alternatively, each sub-signal can be associated with a distinct QoS group. In this case, receivers "tune" into different QoS groups (using signal selection) to build up the overall signal. Both methods are supported in DQM. Receivers and senders interact with QoS groups to determine what the baseline service is and tailor their capability to consume the signal by selecting filter styles and specifying the degree of adaptability sustainable (viz. discrete, continuous; see Sect. 4.2).

4.2 QoS specification API

In the original QoS-A, we designed a *service-contract*-based API which formalized the end-to-end QoS requirements of

the user and the potential degree of service commitment of the provider. In this section, we detail extensions to the *flow specification*, *QoS commitment* and *QoS scaling* clauses of the service contract required to accommodate adaptive multilayer flows. The API presented here is not complete in that there are no primitives given for establishing and renegotiating connections or for manipulating QoS groups. Full details of these aspects are given in [Campbell,94].

Multilayered flows are characterized by three sub-signals in the *flowSpec* – a base layer (BL) and up to two enhancement layers (E1 and E2). Each layer is represented by a frame size and subjective or perceptive QoS as illustrated in Table 1. Based on these characteristics, the MPEG-2 coder [Paek,95; Eleftheriadis,95a] determines approximate bit rate for each sub-layer. In the case of MPEG-2's hybrid scalability, BL would represent the main profile bit-rate requirement (e.g., 0.32 Mbps) for basic quality, E1 would represent the spatial scalability mode bit-rate requirement (e.g., 0.83 Mbps) for enhancement and E2 would represent the SNR scalability mode bit-rate requirement (e.g., 1.85 Mbps) for further enhancement. The remaining *flowSpec* performance parameters for *jitter*, *delay* and *loss* are assumed to be common across sub-signals (i.e., a single layer of a multilayer video flow). The QoS commitment field has been extended to offer an adaptive network service that specifically caters for the needs of scalable audio and video flows in heterogeneous networking environments (see Sect. 6).

The *QoSscalingPolicy* field of the *flowSpec* characterizes the degree of adaptation that a flow can tolerate and still achieve meaningful QoS. The scaling policy consists of clauses that cover *adaptation modes*, *QoS filter styles*, and *event/action* pairings for QoS management purposes. Two types of adaptation mode are supported: *continuous mode*, for applications that can exploit any availability of bandwidth above the base layer and *discrete mode* for applications which can only accept discrete improvement in bandwidth based on a full enhancement (viz. E1, E2).

The QoS scaling policy provides user-selectable QoS adaptation and QoS filtering. While receivers select filter styles to match their capability to consume media at the receiver (from the set of temporal filters), senders select filter styles to shape flows in response to the availability of network resources such as bandwidth and delay (from the set of shaping filters). Network-oriented filters (i.e., selection filters) can be chosen by either senders or receivers.

In addition, both senders and receivers can select periodic performance notifications, including available bandwidth, measured delay, jitter and losses for on-going flows. The *signal* fields in the scaling policy allow the user to specify the interval over which a QoS parameter is to be monitored and the user informed. Multiple signals can be selected depending on application needs.

5 Dynamic QoS management

5.1 Architectural components

Dynamic QoS management spans the QoS-A maintenance and flow management planes. In the QoS maintenance plane, the most important aspect of DQM is QoS adaptation in the end-systems. Based on the receiver-supplied QoS scaling policy, QoS adaptors take remedial action to scale flows, inform the user of a QoS indication and degradation, fine tune resources and initiate complete end-to-end QoS renegotiation based on a new *flowSpec* [Campbell,94]. In the flow management plane, DQM consists of two sub-components: *QoS group management* maintains and advertises QoS groups created by senders for the benefit of potential receivers; and *filter management* [Yeadon,94] instantiates and reconfigures filters in a flow at optimal points in the media path at flow establishment time (when new receivers join QoS groups) or when a new *flowSpec* is given on a QoS renegotiation.

In implementation, each of these architectural modules has well-defined interfaces and methods defined in CORBA IDL. CORBA [OMG,93] runs on the end-systems and in the ATM switches, providing a seamless distributed object-oriented environment throughout the communication system base (see [Aurrecoechea,95] for full details).

5.1.1 Illustrative scenario

DQM can be viewed as operating in three distinct domains:

i) *sender-oriented DQM*, where senders select source filters and adaptation modes and establish flow specifications. The sender-side transport protocol provides periodic bandwidth and delay assessments to the source filters (i.e., DRS or SBR filters) which regulate the source flow. Senders create QoS groups which announce the QoS of the flow to receivers via QoS group management;

ii) *receiver-oriented DQM*, where receivers join QoS groups and select the portion of the signal which matches their QoS capability. Receiver-selected network-based filters propagate through the network and perform source and signal selection. In addition, receiver-based QoS filters (i.e., sync-filter and orch-filter) are instantiated by default unless otherwise directed. These filters are used to smooth and synchronize multiple media. The receiver-side transport protocol provides bandwidth management and produces adaptation signals according to the QoS scaling policy; and

iii) *network-oriented DQM*, which provides an adaptive network service (see Sect. 6) to receivers and senders. Network-level QoS filters (i.e., sub-signal, hierarch and hybrid filters) are instantiated based on user selection and propagated in the network under the control of filter management.

In Fig. 2, a sender at end-system A creates a flow by instantiating a QoS group which announces the characteristics of the flow (viz. layer, frame size, subjective quality) and its adaptation mode. Receivers at end-systems B, C and D join the QoS group. In the example scenario shown, the receivers each "tune" into different parts of the multilayer signal: C takes BL, the main profile (which constitutes a bandwidth of 0.32 Mbps for VHS perceptual QoS), B takes BL and E1 (which constitutes an aggregate bandwidth of 1.15 Mbps for super-VHS perceptual QoS), and D takes the complete signal BL+E1+E2 (which constitutes an aggregate bandwidth of 3 Mbps for laser disc perceptual QoS). In this

```
typedef enum {MPEG1, MPEG2, H261, JPEG}           mediaType;
typedef enum {besteffort, adaptive, guaranteed}   commit;
typedef enum {continuous, discrete}               adaptMode;

typedef enum {
    DRS, SBR, sub_signal, hierarch, hybrid, sync, orch
} filterStyle;

typedef struct {
    adaptMode          adaptation;
    filterStyle        filtering;
    events             adaptEvents;
    actions            newQoS;
    signal             bandwidth;
    signal             loss;
    signal             delay;
    signal             jitter
} QoSscalingPolicy;

typedef struct {
    gid                  flow_id;
    mediaType            media;
    commit               commitment;
    subFlow              BL;
    subFlow              E1;
    subFlow              E2;
    int                  delay;
    int                  loss;
    int                  jitter;
    QoSscalingPolicy     qospolicy
} flowSpec;
```

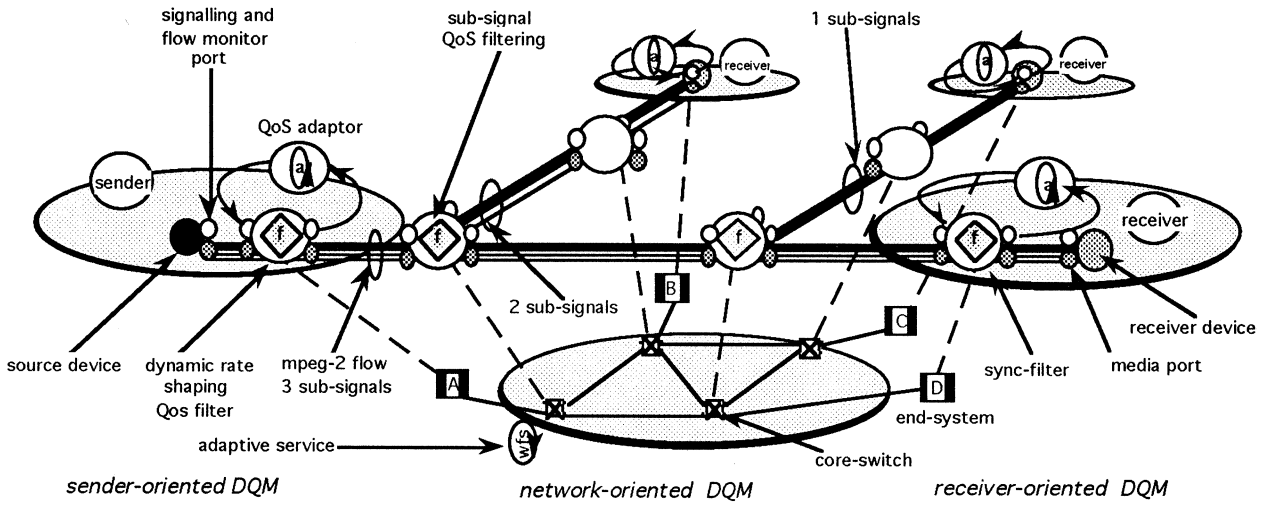


Fig. 2. Dynamic QoS management of scalable flows

example, the complete signal is multiplexed onto a single flow, therefore, sub-signal selection filters are propagated by filter management. Receivers, senders, or any third party or filter management can select, instantiate and modify source, network and receiver-based QoS filters.

5.2 Sender-oriented DQM

Figure 3 illustrates the functions of the sender-side transport protocol supporting dynamic QoS management and the interface to a dynamic rate-shaping filter. Currently, senders can select from two types of shaping filter at the source: dynamic rate-shaping (DRS) and source bit-rate (SBR) QoS filters. Both of these QoS filters manipulate the signal to meet the available bandwidth by keeping the signal meaningful at the receiver. The sender-side transport mechanisms include a QoS adaptor, flow monitor and media scheduler. Bandwidth updates are synchronously received by the flow monitor mechanism from the network as part of the adaptive service (described in Sect. 6). The QoS adaptor is responsible for synchronously informing the source filter of the current bandwidth availability (B_{flow}) and measured delay (D_{flow}), and calculating new schedules and deadlines for transport service data units [Coulson,95]. Media progresses

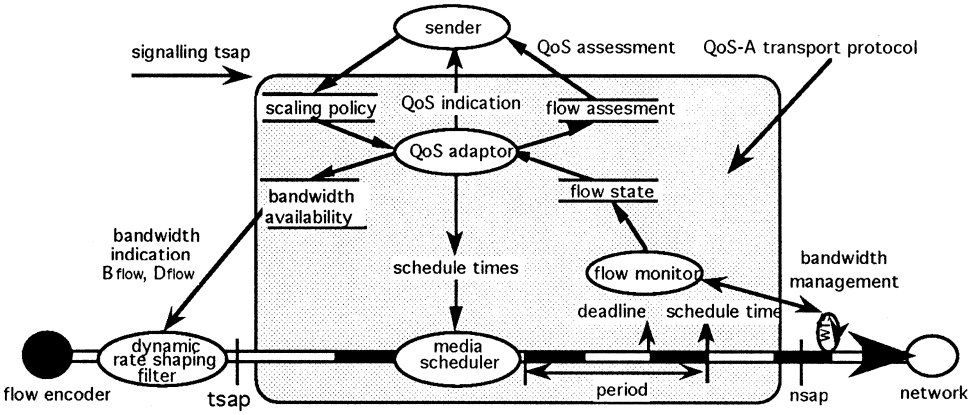


Fig. 3. Sender-side transport QoS mechanisms

from the source filter to the TSAP and is scheduled by the media scheduler to the network at the NSAP based on the calculated deadlines.

The QoS adaptor is also responsible for informing the sending application of the on-going QoS based on options selected in the QoS scaling policy. Informing the application of the current state of the resources associated with a specific flow is key in implementing adaptive applications in end-systems. In this case, the application manages the flow by receiving updates and interacting with the QoS adaptor to adjust the flow (e.g., change adaptation mode from continuous to discrete, request more bandwidth for BL, E1 and E2, or change the characteristics of the source filter, etc.).

5.2.1 The DRS filter

We define rate shaping as an operation which, given an input video bitstream and a set of rate constraints, produces a video bitstream that complies with these constraints. For our purposes, both bitstreams are assumed to meet the same syntax specification, and we also assume that a motion-compensated block-based transform coding scheme is used. This includes both MPEG-1 and MPEG-2, as well as H.261 and so-called "motion" JPEG.

Although a number of techniques have been developed for the rate shaping of *live* sources [Kanakia,93], these cannot be used for the transmission of precompressed material (e.g., in VoD systems). The dynamic rate-shaping filter is interposed between the encoder and the network and ensures that the encoder's output can be perfectly matched to the network's QoS characteristics. The filter does not require interaction with the encoder, and hence is fully applicable to both live and stored video applications.

Because the encoder and the network are decoupled, universal interoperability can be achieved between codecs and networks and also among codecs with different specifications. An attractive aspect is the existence of low-complexity algorithms which allow software-based implementation in high-end computers. In order for rate shaping to be viable, it has to be implementable with ease, while yielding acceptable visual quality. With respect to complexity, the straightforward approach of decoding the video bitstream and recoding it at the target rate would be obviously unacceptable; the delay incurred would also be an important deterrent. Hence, only algorithms of complexity less than that of a cascaded decoder and encoder are of practical interest. These algorithms operate directly in the compressed domain of the video signal, manipulating the bitstream so that rate reduction can be effected. In terms of quality, it should be noted that recoding does not necessarily yield optimal conversion. In fact, since an optimal encoder (in an operational rate-distortion sense) is impractical due to its complexity, recoding can only serve as an indicator of an acceptable quality range. Regular recoding can be lacking in terms of quality with dynamic rate shaping providing significantly superior results.

The rate-shaping operation is depicted in Fig. 4. Of particular interest is the source of the rate constraints $B_{\text{flow}}(t)$. In the simplest of cases, $B_{\text{flow}}(t)$ may be a constant and known *a priori* (e.g., the bandwidth of a circuit-switched connection). It is also possible that $B_{\text{flow}}(t)$ has a well-known statistical characterization (e.g., a policing function). In our approach, $B_{\text{flow}}(t)$ is generated by the adaptive network service.

The objective of a rate-shaping algorithm is to minimize the conversion distortion, i.e.,:

$$\min_{B(t)<T_T(t)} \|y(t) - \hat{y}(t)\| \ .$$

The attainable rate variation ($\hat{B}/B$) is in practice limited, and depends primarily on the number of B pictures of the bitstream. No assumption is made on the rate properties of the input bitstream, which can indeed be arbitrary. There are two fundamental ways to reduce the rate:

i) by modifying the quantized transform coefficients by employing coarser quantization, and
ii) by eliminating transform coefficients.

In general, both schemes could be used to perform rate shaping. Requantization, however, leads to recoding-like algorithms which are not amenable to fast implementation and do not perform as well as selective transmission ones. A selective transmission approach gives rise to a family of different algorithms, that perform optimally under different constraints; for full details see [Eleftheriadis,95b].

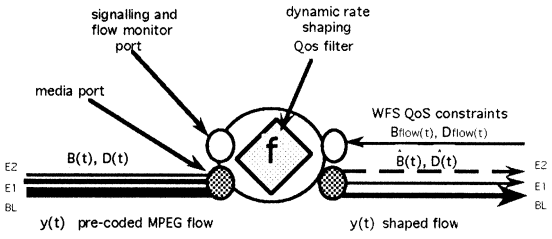


Fig. 4. Dynamic rate-shaping scheme

5.3 Receiver-oriented DQM

QoS adaptors, which are resident in the transport protocol at both senders and receivers, arbitrate between the receiver-specified QoS and the monitored QoS of the on-going flow. In essence, the transport protocol "controls" the progress of the media, while the receiver "monitors and adapts" to the flow based on the flow specification and the scaling policy. When the transport protocol is in monitoring mode [Campbell,94], the flow monitor uses an absolute timing method to determine frame reception times based on timestamps/sample-stamps [Jeffay,92, Jacobson,93, Shenker,93]. The flow monitor, as shown in Fig. 5, updates the flow state to include these measured reception time statistics. Based on these flow statistics, the sync-filter (see Sect. 5.3.3) derives new playout times used by the media scheduler to adjust the playout point of the flows to the decoding delivery device.

QoS mechanisms that intrinsically support such adaptive approaches were first recognized in the late 1970s by Cohen [Cohen,77] as part of research in carrying voice over packet-switched networks. More recently, adaptive QoS mechanisms have been introduced at part of the Internet suite of application-level multimedia tools (e.g., *vat* [Jacobson,93], *ivs* [Turletti,93], and *vic* [McCanne,94]). Vat, which is used for voice conferencing, recreates the timing characteristics of voice flows by having the sender timestamp on-going voice samples. The receiver then uses these timestamps as a basis to reconstruct initial flow, removing any network-induced jitter prior to playout. These multimedia tools are widely used in the Internet today and have proved moderately successful, given the nature of best effort delivery systems (i.e., no resource reservation is made). In the near future, however, an integrated-services Internet [Braden,94] will offer support for flow reservation (e.g., RSVP [Zhang,95]) and new QoS commitments (e.g., predictive QoS [Shenker,95]) which are more suitable for continuous media delivery.

Receiver-oriented adaptation can be divided into a number of receiver-side transport functions (i.e., *bandwidth management, late frame management* and *delay-jitter management*) which are described in the following sub-sections (please refer to Fig. 5). We argue here that these adaptive QoS mechanisms are inherently part of the transport protocol and not, as in the case of vat, ivs and vic, part of the application domain itself.

5.3.1 Bandwidth management

Bandwidth management receives bandwidth indications in the control message portion of the TSDU (or in separate control messages) and adapts the receiver appropriately. The adaptive service, built on the notion of weighted fair-share resource allocation, (see Sect. 6.1) periodically informs the receiver that more bandwidth is available or announces that the flow is being throttled back. Bandwidth management only covers the enhancement signals of multiresolution flows. The base layer is not included, since resources are guaranteed to the base layer. The announcement of available bandwidth on a flow allows the receiver to take either a full or partial enhancement layer. The choice depends on whether the flow is in continuous or discrete adaptation mode.

5.3.2 Late-frame management

Late-frame management monitors late arrivals in relation to the loss metric and the current playout times and takes appropriate action to trade off timeliness and loss. Packets that arrive after their expected playout points are discarded by the media scheduler and the late-packet metrics in the playout statistics are updated. The media scheduler is based on a split-level scheduler architecture [Coulson,95], which provides hard deadline guarantees to base layer flows via admission control and best effort deadlines to enhancements layers. Some remedial action may be taken by the QoS adaptor should the loss metric exceed the loss parameter in the flow specification. If the QoS adaptor determines that too many packet losses have occurred over an era, it pushes out the playout time to counteract the late state of packets from the network. Similarly, if loss remains well within the prescribed ranges, then the QoS adaptor will automatically and incrementally "pull in" the playout time until loss is detected.

5.3.3 Delay jitter management

Our transport protocol utilizes *sync-filters* for delay-jitter management by calculating the playout times of flows based on the user-supplied jitter parameter in the flow specification[2]. Sync-filters calculate the mean and variation in the end-to-end delay based on reception times measured by the flow monitor. Sync-filters take the absolute, mean and variation in delay into account when calculating the playout estimate. A smoothing factor based on a linear recursive filtering mechanism characterized by a smoothing constant is used to dampen the movement of the playout adjustment. Intuitively, the playout time needs to be set "far enough" beyond the delay estimate so that only a small fraction of the arriving packets are lost due to late packets. The QoS adaptor trades off late packets versus timeliness based on the delay and loss parameters in the flow specification. The objective of delay-jitter management is to pull in the playout offset, while the objective of late-packet management is not to exceed the loss characterized in the service contract. The QoS adaptation manager moderates between timeliness and loss. Based on these metrics, the adaptation policy can

[2] Temporal filters can also operate on multiple related audio and video flows to provide low-level orchestration management (in conjunction with the orch-filter). These filter types, however, are not discussed further in this paper.

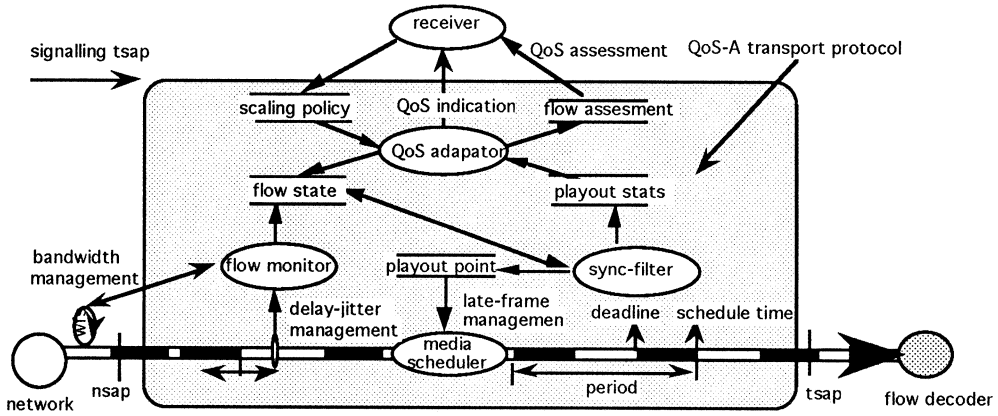


Fig. 5. Receiver-side transport QoS mechanisms

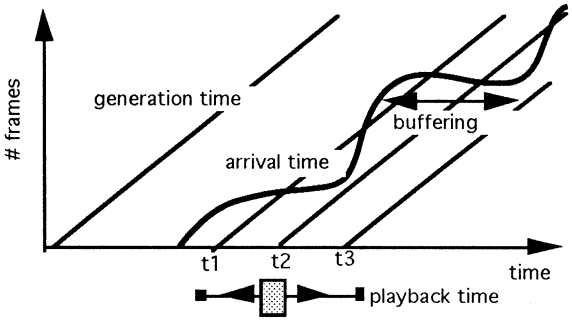


Fig. 6. Sync-filter: timeliness and packet loss regulation

adjust the damping factor and acceptable ranges over which the playout point can operate.

Figure 6 [Zhang,94] shows packets arriving at the receiving end-system. Each packet includes a timestamp used in calculating the flow statistics for delay-jitter management. Selection of the playout point is important: an aggressive playout time which favors timeliness (such as t_1) will result in a large number of late packets. In contrast, a conservative playout point (such as t_3) will be less responsive and timely but will result in no identifiable packet loss. In the DQM scheme, late packets are the same as lost packets, and therefore the loss parameter in the flow specification moderates. An optimum playout schedule is represented by t_2 in the diagram. Continuous media delivery benefits from timely delivery with the exception of some packet loss, which may be deemed acceptable to the receiver in media perception terms.

6 Adaptive network service

The adaptive network service provides "hard" guarantees to the base layer (BL) of a multilayer flow and "weighted fair-share" (WFS) guarantees to each of the enhancement layers (E1 and E2). To achieve this, the base layer undergoes a full end-to-end admission control test [Coulson,95]. On the other hand, enhancement layers are admitted without any such test, but must compete for residual bandwidth among all other adaptive flows. Enhancement layers are rate-controlled based on explicit feed back about the current state of the ongoing flow and the availability of residual bandwidth.

6.1 Weighted fair-share resource partitioning

Both end-system and network communication resources are partitioned between the deterministic and adaptive service commitment classes. This is achieved by creating and maintaining "firewall" capacity regions for each class. Resources reserved for each class, but not currently in use, can be "borrowed" by the best effort service class on condition of preemption [Coulson,95]. The adaptive service capacity region (called the available capacity region and denoted by B_{avail}) is further sub-divided into two regions: i) guaranteed capacity region (B_{guar}), which is used to guarantee all base-rate layer flow requirements; and ii) residual capacity region (B_{resid}), which is used to accommodate all enhancement rates where competing flows share the residual bandwidth.

Three goals motivate our adaptive service design. The first goal is to admit as many base-layer (BL) sub-signals as possible. As more base layers are admitted the guaranteed capacity region B_{guar} grows to meet the hard guarantees for all base signals. In contrast, the residual capacity region B_{resid} shrinks as enhancement layers compete for diminishing residual bandwidth resources. The following invariants must be maintained at each end-system and switch:

$$B_{\text{avail}} = B_{\text{guar}} + B_{\text{resid}}, \quad \text{and} \sum_{i=1}^{N} BL_{(i)} \leq B_{\text{avail}} \ .$$

Our second goal is to share [Steenstrup,94; Tokuda,92] the residual capacity B_{resid} among competing enhancement sub-signals based on a flow-specific *weighting factor*, W, which allocates residual bandwidth in proportion to the range of bandwidth requested that, in turn, is related to the range of perceptual QoS acceptable to the user. In DQM, residual resources are allocated based on the range of bandwidth requirements specified by the users (i.e., BL.. BL+E1+E2 is the range of bandwidth required, e.g., from 0.32 Mbps to 3 Mbps for the hybrid scalable MPEG-2 flow in Table 1). As a result, as resources become available, each flow experiences the same "percentage increase" in the perceptible QoS. We call this *weighted fair-share* (WFS). W is calculated for each flow as the ratio of a flow's perceptual QoS range to the sum of all perceptual QoS ranges.

$$W_i = (BL_i + E1_i + E2_i) \,/\, \sum_{j=1}^{N} \left(BL_j + E1_j + E2_j\right) \ .$$

All residual resources B_{resid} are allocated in proportion to the W metric. Using this factor, we calculate the proportion of residual bandwidth allocated to a flow to be $B_{wfs}(i) = W(i) . B_{resid}$ and the proportion of the available bandwidth allocated to be $B_{flow}(i) = B_{wfs}(i) + BL(i)$.

Our third and final goal is to adapt flows both discretely and continuously based on the adaptation mode. In the discrete mode, no residual bandwidth is allocated by the WFS mechanism, unless a complete enhancement can be accommodated (i.e., $B_{wfs}(i) = E1(i)|E1(i) + E2(i)$, e.g., 0.83 Mbps or 2.68 Mbps from Table 1). In continuous mode, any increment of residual bandwidth Bwfs(i) can be utilized (i.e., $0 < B_{wfs}(i) \leq E1(i) + E2(i)$, e.g., from 0 to 2.68 Mbps from Table 1).

6.2 Rate control scheme

We build on the rate-based scheme described in [Campbell,94], where the QoS-A transport protocol at the receiver measures the bandwidth, delay, jitter and loss over an interval, which we call an "era". An era is simply defined as the reciprocal of the frame rate in the flow specification (e.g., for a frame rate of 24 frames per second as shown in Table 1, the interval era is approximately 42 ms). The receiver-side transport protocol periodically informs the sender side about the currently available bandwidth and the measured delay, loss and jitter. This information is used by the source or *virtual source*[3] to calculate the rate used over the next interval. The reported rate is temporally correlated with the on-going flow. An important result in [Kanakia,93] shows that variable-rate encoders can track QoS variations as long as feedback is available within four frame times or less. This feedback is used by the dynamic rate-shaping filter and network-based filters to control the data generation rate of the video or the selection of the signal, respectively. In the case of dynamic rate shaping, the rate is adjusted, while keeping the perceptual quality of the video flow meaningful to the user.

Based on the concept of eras, control messages are forwarded from the receiver-side transport protocol to either virtual source or the source-side transport protocol using reverse path forwarding. A core-switch [Ballardie,93] where flows are filtered is always considered to be a virtual source for one or more receivers; for full details see [Aurrecoechea,95; Campbell,95]. The WFS mechanism updates the advertized rate as the control messages traverse the switches on the reverse path towards the source or virtual source. Therefore, any switch can adjust the flow's advertized rate before the source or virtual source receives the rate-based control message. The source-side transport protocol hands the measured delay and aggregate bandwidth off (B_{flow}) to the dynamic rate-shaping filter.

DQM maintains the flow state at each end-system and switch that a flow traverses. Flow state is updated by the WFS algorithm and the rate-based flow control mechanism and comprises:

i) *capacity* (viz. B_{avail}, B_{guar}, B_{resid});
ii) *policy* (viz. filterStyle, adaptMode);
iii) *flowSpec* (viz. BL, E1, E2);
iv) *WFS share* (viz., B_{flow}, B_{wfs}, W).

The end-systems hold an expanded state tuple for measured delay, loss and jitter metrics. An admission control test is conducted at each end-system and switch on route to the core for the base layer signal. This test simply determines whether there is sufficient bandwidth available to guarantee the base layer BL, given the current network load:

$$\sum_{j=1}^{N} BL_{(j)} \leq B_{avail} .$$

If the admission control test is successful, WFS determines the additional percentage of the residual bandwidth made available (B_{wfs}) to meet any enhancement requirements in the *flowSpec* :

$$B_{wfs(i)} = W_{fact(i)}.(B_{avail} - \sum_{j=1}^{N} BL_{(j)})$$

The WFS rate computation mechanism causes new B_{wfs} rates to be computed for all adaptive enhancement signals that traverse the output link of a switch. Switches are typically non-blocking, which means the critical resources are the output links; however, our scheme can be generalized to other switch architectures [Coulson,95].

6.3 Network filtering

Currently, our scheme supports two types of selection filters in the network. These are low-complexity and computationally simple filters for selecting sub-signals. Selection filters do not transform the structure of the internal stream; they have no knowledge of the format of the encoded flow above differentiating between BL, E1 and E2 sub-signals. The two basic types of selection filter used are:

i) *sub-signal filters*: these manipulate base and enhancement layers of multilayer video multiplexed on a single flow. The definition of sub-signals is broad; a flow may be comprised of an anchor and scalable extensions or the I and P pictures of MPEG-2's simple profile or the individual hybrid scalable profile. Sub-signal filters are installed in switches when a receiver joins an on-going flow; and
ii) *hierarchical filters*: these manipulate base and enhancement layers, which are transmitted and received on independent flows in a non-multiplexed fashion. In functional terms, sub-signal and hierarchical filters can be considered to be equivalent in some cases. In sub-signal filtering, one flow characterizes the complete signal, and in hierarchical filtering a set of flows characterize the complete signal.

In addition, *hybrid filters* combine the characteristics of sub-signal and hierarchical filtering techniques to meet the needs of complex sub-signal selection. Hierarchical filters, for example, allow the BL, E1 and E2 to be carried over distinct flows and the user can tune into each sub-signal as required.

[3] We use the term *virtual source* to represent a network switch that modifies the source flow via filtering.

As an example, the base and enhancement layers of the hybrid scalable MPEG-2 flow are each made up of I, P and B pictures at each layer i.e., BL (I,P,B), E1 (I,P,B) and E2 (I,P,B). Using hybrid filters, the receiver can join the BL QoS group for the main profile and the E1 QoS group for the spatial enhancement and then select sub-signals within each profile as required (e.g., the I and P pictures of the BL).

7 Conclusion

At Lancaster University, heterogeneity issues present in applications, communications systems and networks are being investigated. Resolving heterogeneous QoS demands in networked multimedia systems is a particularly acute problem that we are addressing within the framework of our QoS-A. As part of that work, we have described a scheme for the dynamic management of multilayer flows in heterogeneous multimedia and multicast networking environments. Dynamic QoS management manipulates and adapts hierarchically coded flows at the end-systems and in the network using a set of scaling objects. The approach is based on three basic concepts: the scalable profiles of the MPEG-2 standard that can provide discrete adaptation, dynamic rate-shaping algorithms for compressed digital video that provide continuous adaptation, and the weighted fair-share service for adaptive flows. At the present time, DQM has been partially implemented at Lancaster University. The experimental infrastructure at Lancaster is based on Pentium machines running a multimedia enhanced Chorus micro-kernel [Coulson,95] and Linux [Campbell,95] and connected by programmable Olivetti Research Limited 4x4 ATM switches. The experimental results from this phase of the work are detailed in [Campbell,95] and summaries found in [Campbell,96a]. This work is being carried out collaboratively with Columbia University in the USA. At Columbia, we are currently using CORBA [Lazar,94] to propagate selection filters in the network using ASX200, NEC and ATML switches. Another thrust of our work at Columbia is the investigation of the suitability of our transport system, adaptation protocol and QoS filtering techniques to the *Wireless ATM (WATM)* environment [Campbell,96b]. We are experimenting with the use of QoS adaptive multimedia application, mobile ATM signalling protocols and our adaptive transport system to provide QoS-controlled mobility in WATM domains.

Acknowledgements. The authors would like to thank members of the Multimedia Projects Group at Lancaster and the ISO QoS Working Group for the many enlightening discussions on end-to-end QoS. In particular, the authors are very grateful to the Alexandros Eleftheriadis and Cristina Aurrecoechea for their thoughtful comments and suggestions on this work. The research on dynamic rate shaping was the subject of Alexandros Eleftheriadis' Thesis at Columbia University. The QoS-A project is funded as part of the UK SERC Specially Promoted Programme in Integrated Multiservice Communication Networks (GR/H77194) in co-operation with GDC (formally Netcomm Ltd). Andrew Campbell wishes to thank the EPSRC for their kind support while he was visiting the Center for Telecommunications Research, Columbia University.

References

[Aurrecoechea,95] Aurrecoechea C, Campbell A, Hauw L, Hadama H (1995) A Model for Multicast for the Binding Architecture. Technical Report, Center for Telecommunications Research, Columbia University, USA

[Ballardie,93] Ballardie T, Francis P, Crowcroft J (1993) Core Based Tree (CBT): An Architecture for Scalable Inter-Domain Multicast Routing. Proc. ACM SIGCOMM '93, San Francisco, USA

[Barden,94] Braden R, Clark D, Shenker S (1994) Integrated Services in the Internet Architecture: an Overview. Request for Comments, RFC-1633

[Campbell,92] Campbell AT, Coulson G, Garcia F, Hutchison D (1992) A Continuous Media Transport and Orchestration Service. Proc. ACM SIGCOMM '92, Baltimore, Maryland, USA

[Campbell,94] Campbell AT, Coulson G, Hutchison D (1994) A Quality of Service Architecture. ACM Computer Communications Review

[Campbell,95] Campbell AT (1995) A Quality of Service Architecture. Ph.D. Thesis, Lancaster University, England, http::/www.ctr.columbia.edu/ campbell

[Campbell,96a] Campbell AT, Coulson G (1996a) Transport QoS Programmability. Proc. ACM Multimedia '96, Boston, USA

[Campbell,96b] Campbell AT (1996b) Towards End-to-End Programmability for QoS Controlled Mobility in ATM Networks and their Wireless Extension. Proc. 3rd International Workshop on Mobile Communications, Princeton, USA

[Clark,84] Clark D, Tennenhouse DL (1984) Architectural Consideration for a New Generation of Protocols. Proc. ACM SIGCOMM '90, Philadelphia, USA

[Cohen,77] Cohen D (1977) Issues in Transit Packetized Voice Communication. Proc. Fifth Data Communications Symposium, Snowbrid, USA

[Coulson,93] Coulson G, Blair G (1993) Micro-kernel Support for Continuous Media in Distributed Systems. Computer Networks and ISDN System

[Coulson,95] Coulson G, Campbell A, Robin P (1995) 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

[Delgrossi,93] Delgrossi L, Halstrinck C, Henhmann DB, Herrtwich RG, Krone J, Sandvoss C, Vogt C (1993) Media Scaling for Audio-visual Communication with the Heidelberg Transport System. Proc ACM Multimedia'93 Anaheim, USA

[Eleftheriadis,95a] Eleftheriadis A, Anastassiou D (1995) Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Code Digital Video. Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire, USA

[Eleftheriadis,95b] Eleftheriadis A (1995b) Dynamic Rate Shaping of Compressed Digital Video. Ph.D. Thesis, Columbia University, USA

[H.262.94] H.262 (1994) Information Technology - Generic Coding of Moving Pictures and Associated Audio. Committee Draft, ISO/IEC 13818-2, International Standards Organisation, UK, March 1994

[Hoffman,93] Hoffman D, Speer M, Fernando G (1993) Network Support for Dynamically Scaled Multimedia Data Streams. Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, UK

[Jacobson,93] Jacobson V (1993) "VAT: Visual Audio Tool. vat manual pages

[Jeffay,92] Jeffay K, Stone DL, Talley T, Smith FD (1992) Adaptive, Best Effort Delivery of Digital Audio and Video Across Packet-Switched Networks. Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, USA

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

[Lasar,90] Lazar AA, Temple A; Gidron (1990) An Architecture for Integrated Networks that Guarantees Quality of Service. International Journal of Digital and Analog Communications Systems, Vol. 3, No. 2

[Lazar,94] Lazar AA, Bhonsle S, Lim KS (1994) A Binding Architecture for Multimedia Networks. Proceedings of COST-237 Conference on Multimedia Transport and Teleservices, Vienna, Austria

[McCanne,94] McCanne S, Jacobson V (1994) VIC: Video Conference U.C. Berkeley and Lawrence Berkeley Laboratory. Software available via ftp://ftp.ee.lbl.gov/conferencing/vic

[OGM,93] OMG (1993) The Common Object Request Broker: Architecture & Specification, Rev 1.3, December 1993

[Peak,95] Paek S, Bocheck P, Chang S-F (1995) Scalable MPEG-2 Video Servers with Heterogeneous QoS on Parallel Disk Arrays. Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire, USA

[Pasquale,93] Pasquale G, Polyzos E, Anderson E, Kompella V (1993) Fitter Propagation in Dissemination Trees: Trading Off Bandwidth and Processing in Continuos Media Networks. Proc. Forth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, UK

[Pegler,95] Pegler D, Hutchison D, Lougher P, Shepherd D (1995) A Scalable Multimedia Storage Hierarchy. Technical Report, MPG-01-95, Lancaster University, England

[Saltzer,84] Saltzer J, Reed D, Clark D (1984) End-to-end Arguments in Systems Design. ACM Trans. on Computer Systems, Vol. 2, No. 4.

[Shacham,92] Shacham N (1992) Multipoint Communication by Hierarchically Encoded Data. Proc. IEEE INFOCOM'92, Florence, Italy, Vol.3, pp. 2107-2114

[Shenker, 93] Shenker S, Clark D, Zhang L (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

[Shenker,95] Shenker S, Partridge C (1995) Specification of Predictive Quality of Service. Working Draft, draft-ietf-intserv-predictive-svc-00.txt

[Steenstrup,92] Steenstrup M (1992) Fair Share for Resource Allocation. pre-print

[Tennenhouse,90] Tennenhouse DL, (1990) "Layered Multiplexing Considered Harmful. Protocols for High-Speed Networks, Elsevier Science Publishers (North-Holland)

[Tokuda,92] Tokuda H, Tobe Y, Chou STC, Moura JMF (1992) Continuous Media Communication with Dynamic QoS Control Using ARTS with an FDDI Network. Proc. ACM SIGCOMM '92, Baltimore, USA

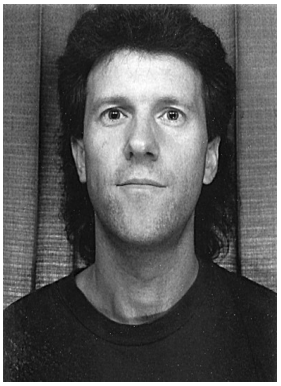
[Turletti,93] Turletti T (1993) A H.261 Software Codec for Videoconferencing over the Internet. INRIA Technical Report 1834, France

[Yeadon,94] Yeadon N, Garcia F, Campbell A, Hutchison D (1994) QoS Adaptation and Flow Filtering in ATM Networks. 2nd International Workshop on Advanced Teleservices and High Speed Communication Architectures, Heidelberg, Germany

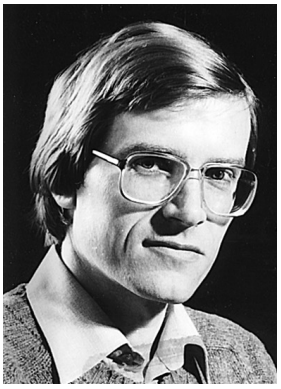
[Zhang,94] Zhang L (1994), Symposium on Multimedia Networking, Columbia University, USA

[Zhang,95] Zhang L, et. al. (1995) RSVP Functional Specification. Working Draft, draft-ietf-rsvp-spec-07.ps

ANDREW T. CAMPBELL joined the E.E. faculty at Columbia as an Assistant Professor in January 1996 from Lancaster University, where he had conducted research in multimedia communications as a British Telecom Research Lecturer. Before joining Lancaster University, Dr. Campbell worked for 10 years in industry, focusing on the design and development of network operating systems, communication protocols for packet-switched and local area networks, and tactical wireless commucation systems. Dr. Campbell is deeply interested in the confluence of quality of serviece and networked multimedia systems research. He is a member of the COMET Group at Columbia's Center for Telecommunications Research. The COMET Group's mission is to build open QoS programmable multimedia networks for the 21st Century. He is currently a co-chair of the IFIP Fifth International Workshop on Quality of Service (IWQoS'97).



GEOFF COULSON received a first-class honours degree in computer science and PhD in the area of systems support for multimedia applications from the University of Lancaster, UK. He is currently a lecturer at Lancaster and is managing a number of research projects. These include the SUMO project, which is investigating operating system support for continuous-media communications and application support, and WAND, a European-Commission-funded collaborative project, which is developing a mobile ATM demonstrator. His research interests are distributed systems architectures, multimedia communications and operating sytem support for continuous media. He has organised and served on the program committees of numerous conferences and workshops in his area.



DAVID HUTCHISON is Professor in Computing at Lancaster University and has been actively involved in research in local area network architecture and distributed systems for the past 14 years, first at Strathclyde University, then (since 1984) at Lancaster. He has completed many UK EPSRC-supported research contracts (including Alvey, ACME and Teaching Company projects) and published over 80 papers and a book in the area. The main theme of his current research is architecture, services and protocols for distributed multimedia systems. He is involved in several recently started projects in these areas, in which an integrating theme is quality of service for multimedia communications. Professor Hutchison is Honorary Editor of the recently launched international Distributed Systems Engineering Journal, published by the IEE, BCS and IoP. He is a programme committee member for many international workshops and conferences.