END-TO-END QoS MANAGEMENT FOR ADAPTIVE VIDEO FLOWS

Andrew Campbell, Alexandros Eleftheriadis and Cristina Aurrecoechea

Center for Telecommunications Research,
Columbia University
New York, NY 10027-6699.
{campbell, elef, cris}@ctr.columbia.edu

INTRODUCTION

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 introduce *dynamic QoS management (DQM)* for the control and management of multi-layer coded flows operating in heterogeneous multimedia networking environments. Two key techniques are proposed: i) an end-to-end *dynamic rate shaping* scheme which adapts the rate of MPEG-coded [1] flows to the available network resources while minimising the distortion observed at the receiver; and ii) 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*. We also discuss a number of types of *media scaling object*[1] which are used to manage and control end-to-end QoS. These include *QoS filters* which manipulate multi-layer coded flows [2] 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.

MEDIA SCALING OBJECTS AND QoS-BASED API

In this section, we introduce a set of *media scaling objects* used to manipulate flows as they progress through the communications system. These comprise QoS adaptors, QoS filters and QoS groups:

Media Scaling Objects

QoS adaptors are used in conjunction with flow monitoring [8] function to ensure that the user and provider QoS specified in the service contract are actually maintained. In this role QoS adaptors are seen as quality of service arbiters between the user and network. QoS adaptors scale flows at the end-systems based on a user supplied QoS scaling policy (see QoS specification API section) and the measured performance of on-going flows.

---

1 *Media scaling* is a general term, first proposed by [3], we use to refer to the dynamic manipulation of media flows as they pass through a communications channel.

QoS filters manipulate multi-layer coded flows [4] [5] [6] 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 section on sender-oriented DQM;

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 section on network filtering;

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 section on receiver-oriented DQM).

Before potential senders and receivers can communicate they must first join a QoS group [7]. The concept of a QoS group is used to associate a baseline QoS capability to a particular flow. All sub-signals of a multi-layer stream can be mapped into a single flow and multicast to multiple receivers [1]. Then, each receiver can select to take either the complete signal advertised 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; that is whether flows are discrete or continuous adaptive [3].

QoS-Based API

In [8] we designed a service contract based API which formalised the end-to-end QoS requirements of the user and the potential degree of service commitment of the provider. An important aspect of our API is that it shield the application from the complexity of QoS management and control [29]. In this section, we detail extensions to the *flow specification*, *QoS commitment* and *QoS scaling* clauses of the service contract required to accommodate adaptive multi-layer 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 [8].

```
typedef enum {MPEG1, MPEG2, H261, JPEG}            mediaType;
typedef enum {besteffort, adaptive, deterministic} 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;
```

Multi-layered flows are characterised by three sub-signals in the *flowSpec* flow specification: 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 [9]. Based on these characteristics, the MPEG-2 coder [9], [10] determines approximate bit rate for each sub-layer. In the case of MPEG-2's hybrid scalability [9], 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 flow specification performance parameters for *jitter*, *delay* and *loss* are assumed to be common across the all sub-signals (i.e. a single layer of a multi-layer 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 adaptive network service section).

The *QoSscalingPolicy* field of the *flowSpec* characterises 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, senders and receivers can both 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.

DQM ARCHITECTURAL COMPONENTS

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* [8]. DQM consists of two sub-components: *QoS group management* maintains and advertises QoS groups created by senders for the benefit of potential receivers; *filter management* [11] 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 [12] runs on the end-systems and in the ATM switches, providing a seamless object oriented environment throughout the communication system base (see [7] for full details).

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 synchronise multiple media. The receiver-side transport protocol provides bandwidth management and produces adaptation signals according to the QoS scaling policy;

iii) *network-oriented DQM*, which provides an adaptive network service (see later) 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 Figure 1 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 multi-layer 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 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.

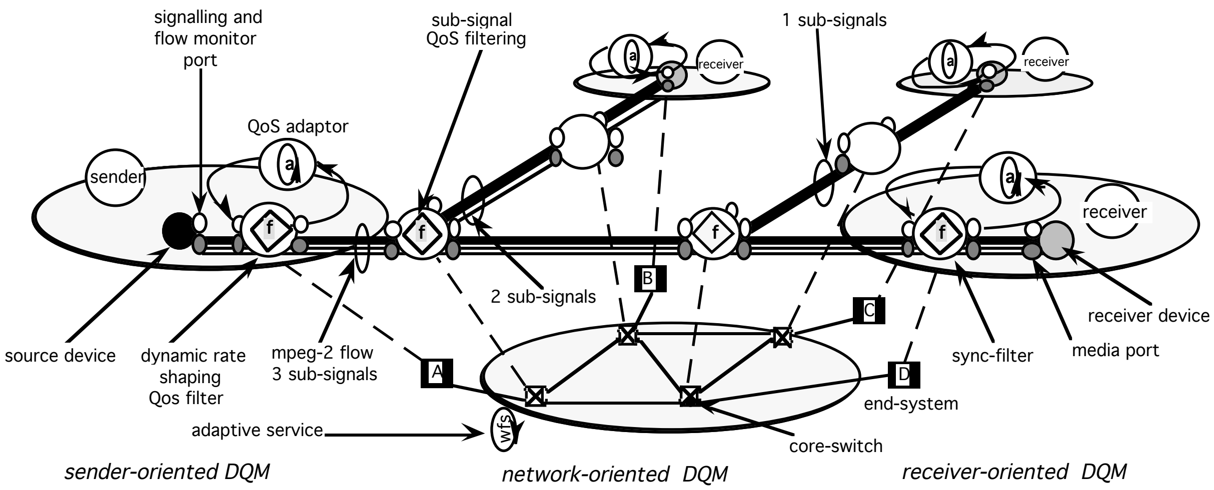

Figure 1: Dynamic QoS management of scalable flows

Sender-Oriented DQM. Figure 2 shows 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 bite 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 includes 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 a later section). 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 [13]. Media progresses 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.

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 [14] these cannot be used for the transmission of pre-compressed 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 quality of service characteristics. The filter does not require interaction with the encoder and hence is fully applicable to both live and stored video applications.

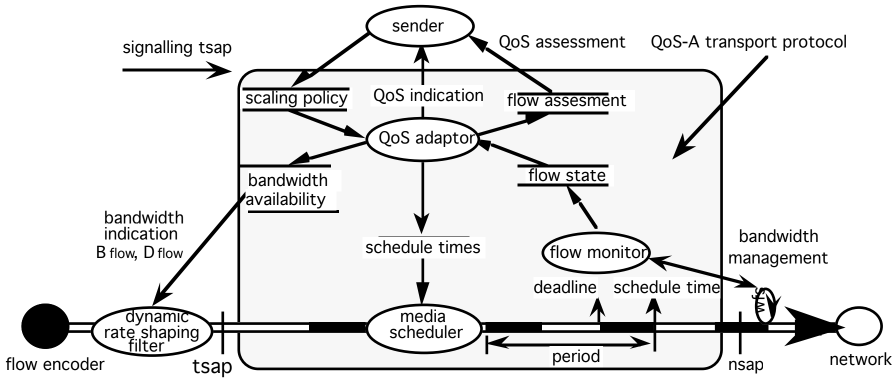

Figure 2: Sender-side transport QoS mechanisms

Because the encoder and the network are decoupled, universal interoperability can be achieved both 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 reasonable 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. In fact, regular recoding can be quite lacking in terms of quality, with dynamic rate shaping providing significantly superior results.

The rate shaping operation is depicted in Figure 3. Of particular interest is the source of the rate constraints $B_{flow}(t)$. In the simplest of cases, $B_{flow}(t)$ may be just a constant and known a priori (e.g. the bandwidth of a circuit-switched connection). It is also possible that $B_{flow}(t)$ has a well known statistical characterisation (e.g. a policing function). In our approach $B_{flow}(t)$ is generated by the adaptive network service.

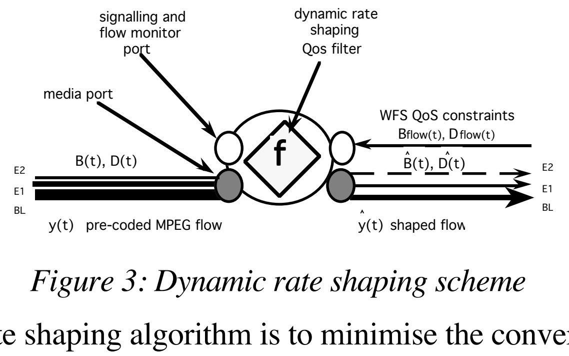

Figure 3: Dynamic rate shaping scheme

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

$$\min_{B(t)<B_T(t)} \left\| y(t) - \hat{y}(t) \right\|$$

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 quantised transform coefficients by employing coarser quantisation, and ii) by eliminating transform coefficients. In general, both schemes could be used to perform rate shaping; requantisation, 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 [15].

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 [8] the flow monitor uses an absolute timing method to determine frame receptions times based on timestamps/sample-stamps [16], [17], [18]. The flow monitor, as shown in Figure 4, updates the flow state to include these measured reception times statistics. Based on these flow statistics, the sync-filter (see section on delay jitter management) derives new playout times used by the media scheduler to adjust the playout point of the flows to the decoding delivery device.

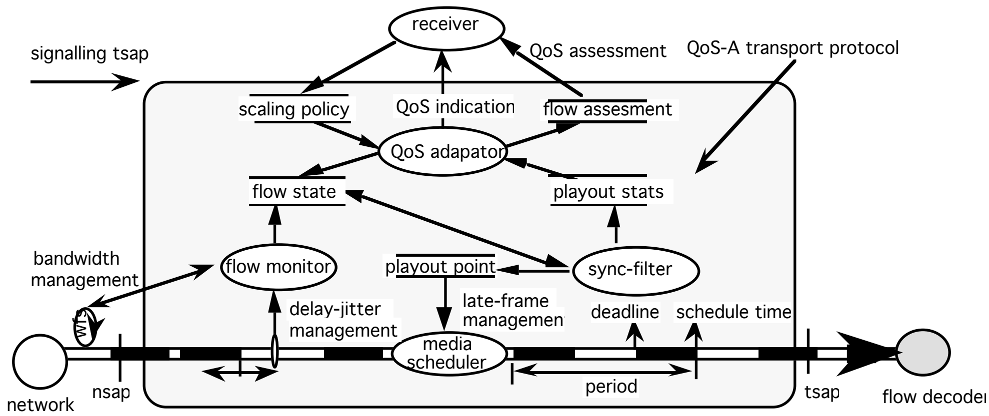

Figure 4: Receiver-side transport QoS mechanisms

QoS mechanisms that intrinsically support such adaptive approaches were first recognised in the late 70's by Cohen [19] 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* [17], *ivs* [20], and *vic* [21]). 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 [22] will offer support for flow reservation (e.g., RSVP [6) and new QoS commitments (e.g., predictive QoS [22]) which are more suitable for continuous media delivery.

Receiver-oriented adaptation can be broken down 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 figure 4). We argue in this paper 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.

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 section on weighted fair share resource partitioning) 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 multi-resolution 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.

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 [13] 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.

Delay Jitter Management. Our transport protocol utilises *sync-filters* for delay-jitter management by calculating the playout times of flows based on the user supplied jitter parameter in the flow specification[1]. 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 characterised 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 characterised in the service contract. The QoS adaptation manager moderates between timeliness and loss. Based on these metrics the adaptation policy can adjust the damping factor, and acceptable ranges over which the playout point can operate.

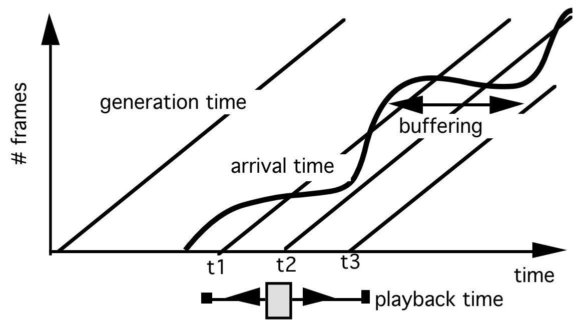

Figure 5: Sync-filter: timeliness and packet loss regulation

Figure 5 [23] 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 favours timeliness (such as t_1) will results 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

---

1 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.

diagram; here, 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.

ADAPTIVE NETWORK SERVICE

The adaptive network service provides "hard" guarantees to the base layer (BL) of a multi-layer 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 [13]. 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 on-going flow and the availability of residual bandwidth.

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 pre-emption [13]. 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:

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

Our second goal is to share [24], [25] 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 [9]). 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.

$$W_{(i)} = (BL_i + E1_i + E2_i) \, / \sum_{j=1}^{N} (BL_j + E1_j + E2_j)$$

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., $B_{wfs}(i) = E1(i) \mid E1(i)+E2(i)$, e.g. 0.83 Mbps or 2.68 Mbps from [9]). In continuous mode any increment of residual bandwidth $B_{wfs}(i)$ can be utilised (i.e. $0 < B_{wfs}(i) \leq E1(i) + E2(i)$, e.g. from 0 to 2.68 Mbps from [9]).

Rate Control Scheme

We build on the rate-based scheme described in [8] where the 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 [9] 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*[1] to calculate the rate to use over the next interval. The reported rate is temporally correlated with the on-going flow. An important result in [14] 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 [26] where flows are filtered is always considered to be a virtual source for one or more receivers; for full details see [7]. The WFS mechanism updates the advertised rate as the control messages traverse the switches on the reserve path to the source or virtual source. Therefore any switch can adjust the flow's advertised 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 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 share 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 generalised to other switch architectures [13].

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, i.e. 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 section filter used are:

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

i) *sub-signal filters*: these manipulate base and enhancement layers of multi-layer video multiplexed on a single flow. The definition of sub-signals is kept general here; 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;

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 characterises the complete signal and in hierarchical-filtering a set of flows characterise 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. For example hierarchical filters allow the BL, E1 and E2 to be carried over distinct flows, and the user can accordingly tune into each sub-signal as required. As an example, the base and enhancement layers of the hybrid scalable MPEG-2 flow are each in turn 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).

CONCLUSION

At Lancaster University we are investigating heterogeneity issues present in applications, communications systems and networks. Resolving heterogeneous QoS demands in networked multimedia systems is a particularly acute problem that we are addressing within the framework of our Quality of Service Architecture (QoS-A) [8]. As part of that work we have described a scheme for the dynamic management of multi-layer 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 is being implemented at Lancaster University. The experimental infrastructure at Lancaster is based on 80486 machines running a multimedia enhanced Chorus micro-kernel [13] and connected by programmable Olivetti Research Limited 4x4 ATM switches. This work is being carried out collaboratively with Columbia University in the US. At Columbia we are currently using CORBA [27] to propagate selection filters in the network using ASX200 switches. In addition to the implementation we are conducting an extensive simulation study into the feasibility of the adaptive network service for large scale use, and investigating the feasibility of extending the adaptive service concept into our enhance Chorus micro-kernel itself [13]. In other QoS related research at Lancaster we are developing a networked scalable multimedia storage system [28] based on the dynamic allocation of compressed file components within a hierarchy of heterogeneous servers, and specialised filter servers [11] which implement transcoder QoS filters.

REFERENCES

1. 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.

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

3. Delgrossi, L., Halstrinck, C., Henhmann, D.B, Herrtwich R.G, Krone, J., Sandvoss, C., and C. Vogt, (1993), “Media Scaling for Audio-visual Communication with the Heidelberg Transport System”, Proc ACM Multimedia'93 Anaheim, USA.

4. Pasquale, G., Polyzos, E., Anderson, E., and V. Kompella, (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.

5. Hoffman, D., Speer, M. and G. Fernando, (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.

6. Zhang, L., et. al., (1995), “RSVP Functional Specification”, Working Draft, draft-ietf-rsvp-spec-07.ps.

7. Aurrecoechea, C., Campbell, A., Hauw, L. and Hisaya Hadama, (1995), “A Model for Multicast for the Binding Architecture”, Technical Report, Center for Telecommunications Research, Columbia University, USA.

8. Campbell, A., Coulson, G. and Hutchison, D., (1994), “A Quality of Service Architecture”, ACM Computer Communications Review.

9. Paek, S., Bocheck, P., and 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.

10. Eleftheriadis, A., and D. Anastassiou, (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.

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

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

13. Coulson, G., Campbell, A and P. Robin, (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.

14. 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.

15. Eleftheriadis, A., (1995b), “Dynamic Rate Shaping of Compressed Digital Video”, Ph.D. Thesis, Columbia University, USA.

16. Jeffay K., Stone, D.L., Talley, T. and F.D. Smith, (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.

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

18. 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.

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

20. Turletti, T, (1993), "A H.261 Software Codec for Video-conferencing over the Internet", INRIA Technical Report 1834, France.

21. 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

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

23. Zhang, L., (1994), Symposium on Multimedia Networking, Columbia University, USA.

24. Steenstrup, M., (1992), “Fair Share for Resource Allocation”, pre-print.

25. Tokuda, H., Tobe, Y., Chou, S.T.C. and Moura, J.M.F., (1992), “Continuous Media Communication with Dynamic QoS Control Using ARTS with an FDDI Network”, Proc. ACM SIGCOMM ‘92, Baltimore, USA.

26. Ballardie, T., Francis, P. and Jon Crowcroft, (1993), “Core Based Tree (CBT) An Architecture for Scalable Inter-Domain Multicast Routing”, Proc. ACM SIGCOMM ‘93, San Francisco, USA.

27. Lazar, A. A., Bhonsle S., Lim, K.S., (1994) “A Binding Architecture for Multimedia Networks”, Proceedings of COST-237 Conference on Multimedia Transport and Teleservices, Vienna, Austria.

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

29. 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