Multimedia Feedback Control in ATM Local Area Networks

Payal Bindal  
Liam Murphy

Department of Computer Science and Engineering  
Auburn University, AL 36849  
{bindapa,lmurphy}@eng.auburn.edu

Abstract

Multimedia networking refers to the transfer of related audio, video, image and data streams among networked computers. A key problem in multimedia networking is the synchronization of audio and video streams to maintain their timing relationships. In addition, due to the large time-varying resource demands of real-time video, an automatic mechanism is needed to dynamically modify the video source output in order to track changing network traffic levels. We have developed a receiver-initiated feedback control scheme to allow video sources to respond to changing network conditions while maintaining synchronization with their associated audio streams. The receiver informs the sender about the current transmission quality, and the sender responds to this information by adjusting some transmission parameters which temporarily decrease video quality (if the receiver is reporting problems) or increase the video quality (if the receiver is reporting no problems but the quality is less than its target). We present simulation results which indicate that our feedback mechanism can help to maintain audio-video synchronization in Asynchronous Transfer Mode local area networks, as well as reduce the network traffic congestion which leads to glitches in the first place.

Keywords: multimedia networking, VBR MPEG video, Asynchronous Transfer Mode, local area networks, feedback control, discrete event simulation.

1 Introduction

Multimedia traffic consists of related information of several types, such as audio, video, images and text. The transfer of multimedia traffic over a computer network is referred to as multimedia networking. Multimedia information streams can be transmitted as a single combined stream (Figure 1(a)) or as multiple separate streams (Figure 1(b)). For many collaborative and conference...
Full color video has a high natural data rate, on the order of 250 million bits per second (250 Mbps). A small number of such sources could easily overload even a high-speed network. Because of this, video information is *compressed* to much lower rates (e.g. 0.5–5 Mbps). Compression algorithms take advantage of the fact that there is often little change between successive pictures (temporal redundancy), and that most changes within a single picture are gradual rather than abrupt (spatial redundancy). Even with compression, a video source transmits a variable bit rate stream into the network.

A video stream usually has an associated audio stream. Audio has been well-studied and can be transmitted at relatively low bit rates. The traditional audio standard in telephony and ISDN is a constant bit rate of 64 kbps. While compression algorithms for voice–quality audio can reduce this to 16 kbps or lower, the bandwidth savings often do not justify the resulting additional cost and complexity.

An audio or a video stream by itself has timing requirements: there is a maximum tolerable end-to-end delay, as well as a need to play out the stream in a continuous manner at the receiver. Occasional transmission errors can be tolerated because of limitations of the human eye or ear. Therefore it is usually preferable to suffer temporary loss of quality while maintaining a steady flow of data within the maximum delay bound, than to retransmit audio or video packets which contain transmission errors.

In addition to these individual timing requirements, a combined audio and video stream requires synchronization at the receiver to be understandable to the end-user. For example, a video sequence whose audio stream is delayed by 20 milliseconds produces an effect known as “lip sync”. The more skewed the two streams are with respect to each other, the more pronounced the effect, and eventually the intended meaning of the transmission is lost. Even if the streams are transmitted synchronously, variations in network load during transport can result in variable delays and ultimately in a loss of synchronization at the receiver. This is referred to as a *glitch* and is perceived by the end-user as a loss in transmission quality.

There are several options available to a network operator to reduce the possibility of glitches, and to react to glitches when they occur so that synchronization can be recovered. In this paper we focus on a feedback mechanism in which the receiver informs the sender about the current transmission quality: whether the received streams are synchronized and, if not, how much out-of-sync the audio and video streams are. The sender responds to this information by adjusting some transmission parameters in order to temporarily reduce the transmitted bit rate (which decreases video quality) if the receiver is reporting glitches, or to increase the bit rate (and hence the video quality) if the receiver is reporting no glitches but the video is less than the desired quality. By reducing the source bit rate in times of network congestion (when glitches are more likely), our feedback scheme is also useful as a congestion control measure.

The rest of the paper is organized as follows: first, we briefly review the standard encoding method for video (MPEG encoding) and the basics of Asynchronous Transfer Mode (ATM) networks, and outline our modeling assumptions about them. Then we discuss some other multimedia traffic control schemes reported in the literature, before describing our feedback control scheme and the different implementations of it that we investigated. We then present simulation results which indicate that our feedback scheme can help to maintain audio–video synchronization in ATM local area networks, as well as reduce the traffic congestion which leads to synchronization problems in the first place. We conclude with some discussion of our results and possibilities for future work.

## 2 MPEG encoding

The Motion Pictures Expert Group (MPEG) has developed a series of standards for video encoding [6] which have been widely deployed. The basic MPEG encoding scheme includes the following types of video frames:

- **I frame** – this is a “true” picture, meaning it is encoded independently of other pictures;
- **P frame** – this is a “predicted” picture, meaning it specifies the differences from a previous I or P frame rather than being an actual picture.

The receiver uses P frames to calculate what should be displayed to the user. Since P frames are not pictures, but offsets from previous pictures, they require much lower network bandwidth than I frames; this is what makes MPEG video a variable bit rate source. Therefore temporal compression is achieved by transmitting \( I_{ratio} \) P frames between successive I frames, as shown in Figure 2. On the other hand, some picture quality is inevitably lost due to the limited information available in a P frame. Therefore the value of \( I_{ratio} \) should be set to a value which limits the loss of picture quality while still allowing a desirable compression level. In our simulations we set \( I_{ratio} = 7 \).

The frame rate refers to the transmission of frames at the sender, which ideally would also be the rate at which

---

1 There are other types of MPEG frame which we do not consider here.
those frames are played out at the receiver. In practice, since some frames may be delayed or lost, MPEG decoders have some method for ensuring a continuous playout process (e.g. replay the previous P frame, or extrapolation from the last few frames). We assumed a transmitted frame rate of 30 frames per second.

Each frame is decomposed into horizontal strips called slices which are coded independently to minimize error propagation. The number of slices per frame is set in the encoder and decoder, and is constant for the duration of the transmission. In our simulations we assumed 15 slices per frame.

Spatial compression is achieved by quantizing picture information into a number of discrete values rather than a continuous range. The quantization factor Q controls the granularity of the picture: Q=12 represents the highest picture quality, Q=4 represents the lowest tolerable quality, and integer values of Q in this range represent intermediate values of picture quality. In general, higher values of Q require more information to be transmitted than lower values of Q.

[3] explores the MPEG-2 coding set and ATM networks in relation to Video Dial Tone (VDT) networks. It shows how ATM has emerged as the technology of choice for switching and transport of multimedia connections or point-to-point switched video. The MPEG-2 Systems Recommendation that suggest multiplexing video, audio and data into single or multiple streams suitable for transmission or storage are being adopted by the industry.

3 ATM Networks

Asynchronous Transfer Mode (ATM) is a proposed computer networking method which is expected to be the basis for the high-speed communication networks of the future [9]. The ultimate technological goal of these networks is to carry all information services – such as two-way and multi-way voice and video connections, selection and playout from a video library, photograph and still image transfers, or text file transfers – on one integrated network platform. ATM would provide the low-level services which move data from information sources to destinations in such a way that the end-users receive their information services with acceptable quality.

ATM is currently being standardized and some prototype implementations are available, but the standardization process is incomplete and several problems remain to be solved. Perhaps the most important unsolved problem in ATM networks is how a wide range of information services can be simultaneously provided by a single network infrastructure at a reasonable cost to the users. The challenge is to find ways to combine different types of information so that they use the minimum network resources necessary to satisfy the users' performance requirements.

In ATM networks, information transmitted from sender to receiver is divided up into smaller units called cells. These cells travel through the network and are combined by the receiver to reconstruct the original information. Since ATM network resources can be dynamically shared among several connections [9], audio and video cells for the same frame may arrive at different times at the receiver. In order to reconstruct their timing relationship, a synchronization point is provided at the receiver. All cells are buffered until the corresponding audio and video cells are received. If the playout deadline is missed then a glitch has occurred, which requires some action to be taken to regain synchronization.

ATM is intended for use in the wide area as well as the local area. In wide area ATM networks, the delay between a transmission and receiving feedback information about that transmission from a distant receiver complicates any feedback control scheme. Since ATM is expected to be deployed initially in the local area, where feedback delays are much smaller or negligible, we restrict our attention in this paper to ATM local area networks.

4 Multimedia Traffic Control Schemes

Many schemes have been reported in the literature. We cannot claim that our review is comprehensive, but we
briefly outline some representative approaches. Then we introduce our proposed scheme and describe its principal features.

4.1 Previous Approaches

MPEG video source characteristics and the effect of varying the quantization and $I$ ratio parameters on these characteristics was studied in [8]. MPEG encoding was evaluated at various time scales e.g. frame layer, slice layer, block layer. The time scale for the cell generation process for a slice corresponds to the multiplexing layer in ATM networks. [8] showed that the quantization parameter is highly correlated to the number of bits generated per frame. This suggests that the quantization parameter can be used to control a video source's bit rate.

Dynamic bandwidth allocation for stored and real-time video sources was investigated in [10]. The video sources were modeled as variable bit rate MPEG transmitters. A source's MPEG encoder adjusts the quantization level of the source to control the number of bits stored in the sender's buffer for transmission. [10] shows that dynamic bandwidth allocation with this kind of local feedback could provide significant quality gains while reducing the number of bits transmitted. How to relate the feedback parameters to the video quality perceived by the end-user is not clear, however.

Feedback from within the network may be used by a video source to adapt to changes in network conditions caused by an increase or decrease in the number of connections, or by sudden changes in the sending rates of the existing connections. In [5], a predictive control scheme is proposed which uses network feedback to try to maintain each stream's bottleneck queue at a constant level. As in [10], the source quantization is adjusted to vary the amount of traffic transmitted into the network. A subjective measure of perceptual quality, the Mean Opinion Score (MOS), was used to help evaluate the effectiveness of the feedback scheme. [5] showed that the proposed scheme helps in network congestion control by gracefully degrading picture quality. Our proposed scheme is similar in spirit to this scheme, although we derive the feedback signals in a different (and, we believe, simpler) way.

[11] presents the work in standardization of user-oriented, technology-independent measures of telecommunication service quality. Experiments conducted in [11] present some perception-based objective quality measures that are valuable since they are highly correlated with the subjective viewer responses. Though there is ongoing work to optimize the basic electrical measures and their mappings with human perceptions through additional research and objective/subjective parameter correlation experiments, for now subjective tests remain the only viable reference point for video/audio quality assessment.

Bit rate, digitized signal quality and processing delay are the fundamental criteria in digital coding of audio and video [4]. A well-known metric of signal-to-noise ratio (SNR) is appropriate for analog communication links but is a very poor predictor of performance for motion video. [4] shows that the bit error and packet loss probabilities are useful to evaluate MOS (Mean Opinion Score) which is based on subjective measures. This is similar to our idea of using glitches as a performance metric, since glitches measure the number of frames that missed the deadline i.e. an error (this is user-defined error).

A dynamic QoS Management (DQM) scheme is investigated in [2]. It presents sender or network or receiver-oriented DQM. Details of the receiver-oriented DQM reveal a similar concept of late-frame management to ours, which monitors late arrivals in relation to a loss metric and the current playout times and takes appropriate action to tradeoff timeliness and loss. Though this helps in QoS management, the effect of discarding packets on image quality is not clear.

4.2 A New Receiver-Initiated Feedback Scheme

Many video feedback schemes assume that the network provides feedback to the source, e.g. [5]. However the receiver is in the best position to judge the quality of the connection. In [7], a receiver-driven feedback scheme to maintain audio-video synchronization was proposed, using a degrade algorithm to reduce the source's bit rate when glitches occur, and an upgrade algorithm to improve video quality when synchronization is re-established after a glitch.

In this paper we modify the schemes described in [7] by restricting the feedback parameters to those that can be adjusted by current MPEG encoders. These include

- the source quantization level, $Q$;
- the number of $P$ frames actually transmitted by the sender.

In the latter case, the sender can transmit less than $I$ ratio $P$ frames between consecutive $I$ frames, provided the receiver can be signaled to recover (e.g. by replaying the last $P$ frame received). Since this is already a required feature of MPEG decoders, we assume here that it is a feasible control scheme.

For each video connection, time is divided into playout periods, which correspond here to the intervals between successive $I$ frames. The receiver calculates the playout deadlines and monitors the connection for glitches
during a period. If a glitch occurs, the receiver sends a feedback signal to the video source indicating that the video quality should be degraded in an effort to regain synchronization. Similarly if no glitches occur, the receiver sends a signal back to the source indicating that the quality can be upgraded. The upgrade algorithm is asymmetric to the degrade algorithm: it makes more conservative changes, in an effort to avoid immediately returning to a problematic situation. The degrade and upgrade algorithms used to collect the results reported here are fully described in [1].

4.3 Variable Quantization Only

This is the simplest implementation of our feedback control scheme. Only the source quantization was adjusted to vary the video quality. We distinguish two cases: the simple and aggressive schemes. These differed only in the details of the degrade algorithm; the upgrade algorithm was the same in both cases.

In the simple scheme, the sender responded to a degrade signal from the receiver by decreasing Q by 1 (unless Q is already at its lowest value of 4, in which case no change was made). This reduces the transmitted bit rate at the video source.

In the aggressive scheme, the amount by which a video frame missed its playout deadline was taken into account when giving feedback to the sender. Let amount late be the amount by which a frame missed its deadline:

\[
\text{amount late} = \text{frame arrival time} - \text{deadline}
\]

We also define the quantity \( \text{max late} = 2 \times \frac{1}{\text{frame rate}} \). The aggressive degrade algorithm worked as follows:

- if \( \text{amount late} \geq \text{max late} \), then degrade Q to its lowest level of 4;
- if \( \frac{N}{100} \times \text{max late} \leq \text{amount late} < \text{max late} \), then degrade Q by 3, if possible. If this requires the new value of Q to be less than 4, set Q to 4;
- otherwise, degrade Q by 1 (as in the simple scheme).

We investigated two possible values of \( N \) : \( N = 5 \% \) and \( N = 50 \% \).

4.4 Variable Quantization and Buffers

In this variation, we delay each playout deadline by a small amount from its calculated value. This gives a frame a little "extra" time to synchronize its audio and video components, while having a negligible impact on perceived video quality (since the delays in a local area network are small anyway). This scheme could be implemented by providing a playout buffer at the receiver to hold arriving frames until their (modified) playout deadline is reached. We investigated two such artificial delays: one frame time past the calculated deadline, and two frame times past the calculated deadline. These would require a playout buffer capable of holding one and two frames, respectively. Both simple and aggressive degrade algorithms (section 4.3) were investigated for each buffer size.

4.5 Variable Quantization and Modified Transmission

In this implementation, two transmission modes are identified:

- **mode 1**, in which the sender transmits \( L_{\text{ratio}} \) P frames between successive I frames as described in section 2;
- **mode 2**, in which the sender transmits only \( \frac{L_{\text{ratio}}}{2} \) P frames between successive I frames, replacing each suppressed P frame with a control cell informing the receiver that the frame was not transmitted.

Depending on the recovery mechanism at the receiver, mode 2 may or may not seriously affect the perceived video quality, while reducing the amount of data transmitted by the sender in a given playout period.

This scheme can be combined with the simple and aggressive schemes outlined above. Depending on the implementation, either the Q value or the transmission mode can be adjusted first when a glitch occurs, deferring adjustment of the other parameter until the first is exhausted.

We have implemented this type of scheme but do not yet have any simulation results to report.

5 Simulation Models

Simulations were carried out using SES/Workbench [12], a commercial discrete-event simulation tool. SES/Workbench is run on a Sun Sparstation under the Solaris operating system. Various components can be added using an object oriented approach, using C programming constructs. The main components in our model are video sources, ATM switches, synchronization nodes, receivers, feedback nodes and delay nodes (see Figure 3). As events occur in Workbench, resource nodes generate transactions (ATM cells) according to actions defined within the nodes. Workflows are modeled by creating links between nodes and specifying the type of information that can flow on each link.
5.1 Video and Audio Source Models

Studies attempting to characterize MPEG video sources (e.g. [8]) were used to construct an approximate MPEG video source traffic model. Each video frame is composed of 15 slices. ATM cells corresponding to each slice are sent into the network. A special frame-end transaction signals the end of the current video frame being transmitted. For every video frame that the video source transmits, it also sends one audio frame. The video links are restricted to carrying video ATM cells only; data or audio cells are not carried by these links. The source distinguishes between the I and P frames by the number of corresponding ATM cells sent: we model the P frames as having half as many ATM cells as the I frames. After sending each frame, the source checks for a feedback cell from the receiver. When a feedback cell is received, the video source modifies its parameters according to the flags set in the feedback cell.

An audio source was modeled as a constant bit rate 64 kbps stream. Since AAL 1 would most likely be used to transport audio cells in practice, we assumed this corresponds to 6 ATM cells [7].

5.2 Network Model

The ATM network is modeled by simulating ATM switches and transmitting ATM cells over the network. The ATM switches in our model have a single queue with a FIFO (first-in, first-out) queueing strategy. Using AAL 1, each ATM cell contains 47 bytes of user data, where the user data can be either audio, video or data. Delays corresponding to transmission, propagation and queuing time are simulated by the addition of delay nodes. The feedback delay for the LAN is based on a 100 mile one-way link. The cross traffic on the network is simulated by specifying source destination pairs that transmit video, audio and data cells. Each cell has a category and phase associated with it, specifying its source and the type of data that it contains. This information is used by switches to route the cells onto the correct links and eventually to their destinations. The cross-traffic sources emulate MPEG video encoders. However, their traffic is open-loop i.e. no feedback is applied to these source-destination pairs.

6 Simulation Results

The performance of the various implementations of our feedback control scheme was measured in two ways:

- **glitch count**: the percentage of glitched frames in a 20-second simulated MPEG video sequence;
- **Q quality**: the percentage of frames for which the value of Q was at least 8 during the 20-second sequence.

While these are imperfect measures of the perceived video quality – which is a subjective measure depending on the end-user – we believe that they provide some indication of the video quality likely to be perceived at the receiver. The glitch count results for the various feedback schemes are summarized in Table 1, while the Q quality results\(^2\) are summarized in Table 2.

As a comparison, we first let the video sources run "open loop" i.e. without any feedback. For Q=12, the glitch count was over 98%, indicating that our model represents a heavily-loaded network.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Only Q</th>
<th>Only Q with Buffers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 fr. buffer</td>
<td>2 fr. buffer</td>
</tr>
<tr>
<td>Simple Feedback</td>
<td>48.33%</td>
<td>18.06%</td>
</tr>
<tr>
<td>Aggr. Feedback, N = 50%</td>
<td>39.57%</td>
<td>3.51%</td>
</tr>
<tr>
<td>Aggr. Feedback, N = 5%</td>
<td>25.04%</td>
<td>0.17%</td>
</tr>
</tbody>
</table>

Table 1: Glitch counts

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Only Q</th>
<th>Only Q with Buffers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 fr. buffer</td>
<td>2 fr. buffer</td>
</tr>
<tr>
<td>Simple Feedback</td>
<td>51.25%</td>
<td>51.25%</td>
</tr>
<tr>
<td>Aggr. Feedback, N = 50%</td>
<td>49.91%</td>
<td>49.91%</td>
</tr>
<tr>
<td>Aggr. Feedback, N = 5%</td>
<td>44.57%</td>
<td>44.57%</td>
</tr>
</tbody>
</table>

Table 2: Q quality results

It is seen that using the simple feedback scheme with the Only Q algorithm, the glitch count drops to 48% which is a significant improvement over the open loop case at 98%. But this is still a high glitch count for a multimedia transmission. By adding small playout buffers at the receiver, the glitch count is reduced significantly to 18% and 4% for buffer sizes that equal one frame and two frame sizes, respectively. Thus, with a two frame buffer at the receiver, using the simple feedback with the Only Q with Buffers scheme helps reduce the glitch count to 4%, compared to the 98% glitch count for the open loop case.

The aggressive feedback scheme with the Only Q algorithm brings down the glitch count to 39% for a less aggressive setup (N=50%) and to 25% for a more aggressive setup (N=5%). The Only Q with Buffers algorithm when based on an aggressive feedback scheme makes a big impact on the glitch count. It reduces the glitch count to 3.5% for a less aggressive setup with one frame buffer, and 0% with the more aggressive scheme.

---

\(^2\)Note that the Q quality results are identical for a given degrade algorithm regardless of whether or not a playout buffer is used, as expected, since the sender behavior is the same whether the receiver is using a playout buffer or not.
or two frame buffers. Thus the playout buffers at the receiver, coupled with an aggressive scheme, eliminate the glitches.

The picture quality results show that for the simple feedback schemes 51% of the time the sender is sending frames at a Q level of 8 or above. The aggressive schemes reduce the picture quality slightly to 50% for the less aggressive setup and to 44% for the more aggressive setup.

Thus the results show that even a simple degrade algorithm can greatly increase the video quality at the receiver. More aggressive algorithms further decrease the glitch counts without significant reduction in the Q quality. The use of even a small playout buffer at the receiver can essentially eliminate glitches, at the cost of a small increase in the overall delay.

7 Conclusions and Future Work

We have proposed a receiver-initiated feedback control scheme for VBR MPEG video traffic streams, and presented some encouraging simulation results which suggest that it can be useful for multimedia synchronization and dynamic congestion control. In particular, a small playout buffer combined with an aggressive degrade algorithm can essentially eliminate glitches while still maintaining a reasonable video quality at the receiver.

However, several questions about the project so far remain unanswered, and further development is planned to address these questions. For example, the "best" combination of the various implementations discussed above needs to be defined and determined. The simulation model will be refined to more accurately represent MPEG sources and realistic ATM local area networks. More than one source destination pairs that have feedback need to be investigated for a more realistic approach. Also, the cross traffic will be made more complex than its current setup to view how well these algorithms withstand heavier and unpredictable cross traffic. There is also a need to test the algorithms using other test sequences, to verify that the improvement in performance is not an artifact of the 20-second frame sequence used up to now. Any simulation results, no matter how realistic, can only provide circumstantial evidence of improved viewing quality, so as a longer-term goal we intend to use actual video sequences and equipment to experimentally verify our simulation results. The degree to which our feedback scheme is successful in the wide area (and/or how much it can be modified to be successful) is also currently unknown.

Acknowledgement: the authors would like to thank Johnny Lewis for helpful discussions.

References


Figure 3: Our Simulation Model