Inter-stream Synchronization for Multimedia Applications in Wireless-ATM Networks

A. Iera*, A. Modafferi*, A. Molinaro**, R. Agostino*, G. Araniti*

* University of Reggio Calabria, D.I.M.E.T. - Via Graziella (Feo di Vito), 89100 Reggio Calabria, ITALY
** University of Calabria, Dept. D.E.I.S. – Arcavacata di Rende (CS), 87036 Rende, ITALY

Abstract - This paper presents the analysis of some MAC protocols for multimedia applications in Wireless-ATM networks. The novelty of the approach is the delivery of multimedia applications over separate bearers over the radio interface while maintaining the original synchronization degree. The presented techniques are able to guarantee low jitter, high degree of synchronization among different streams of the same user application, as well as to effectively exploit the limited radio spectrum of a wireless air interface.

I. INTRODUCTION

In the last few years a growing interest in the research topic of multimedia communications has been observed. The explosive growth of the Internet, the enormous diffusion of wireless networks and the recent progresses in the field of broadband wireless access networks opened a new scenario for the widespread diffusion of multimedia services and applications [1]. Unfortunately, the cost effective transport of multimedia traffic and applications over wired and (especially) wireless networks is a difficult task for different reasons, related to the inner properties of the transmission medium and to the nature of the multimedia traffic streams.

A multimedia service or application, according to the definition given in [2], is composed of different types of independent media perceived by the user as a single logic entity. From the traffic point of view, a multimedia application generates different media (voice, video and data) contributions. These contributions (or traffic components) are characterised by different profiles and quality requirements (delay, jitter and/or error tolerance). Nevertheless, they show a strong interdependence.

According to some recent trends and recommendations [3], traffic from the different media of a multimedia connection should be transported over separate ad-hoc bearer services (multi-bearer paradigm). Each bearer is carefully designed to match the requirements of a single component media stream. Thereby, the different traffic profiles and bandwidth demands of the components can be properly and fairly managed in order to effectively exploit the system radio resources as well as to satisfy the user requirements. This transport choice obviously brings to a crucial issue, that is the maintenance of the interdependencies among the monomedia components of the same application. This is the inter-stream synchronization (e.g. video and audio traffic in a video-telephony session) problem.

This paper focuses on the analysis of the effects of the multi-bearer transport solution in a Wireless-ATM (W-ATM) network. The reason of the choice of a W-ATM environment is twofold. First, W-ATM is an extension of the fixed ATM technology, which has an “innate” ability in supporting multimedia traffic with different quality of service (QoS) guarantees. Moreover, the medium access control (MAC) layer is responsible for efficient bandwidth allocation. Thus, a well-designed MAC protocol can assure optimal performance and efficient transport of multimedia traffic over the wireless side of the network. In literature, some MAC algorithms have been proposed, which efficiently exploit the radio bandwidth and deal with heterogeneous traffic sources [1]. However, several works consider only monomedia mobile terminals, transmitting a single traffic type. These MACs are optimised to guarantee QoS to unrelated traffic flows; but obviously, synchronisation among the heterogeneous traffic components of different monomedia terminals is not an issue of interest. Nevertheless, the current trend in the terminal design and manufacturing is towards the use of a multimedia terminal transmitting heterogeneous data flows. When in the presence of such multimedia applications, the MAC algorithms need to be re-designed in order to guarantee synchronization among the traffic components which are delivered over different bearers on the radio access interface.

Another key aspect, which has a deep impact on the MAC performance, is the technique used for coding information from the different media components. Digital video and audio coding has, in fact, a significant influence on the viability and the performance of multimedia applications [1]. MPEG codecs are widely used to encode video and audio streams of many multimedia applications. The transport of MPEG traffic streams over W-ATM networks gives rise to interesting research issues. First of all, the performance of some MAC algorithms proposed in literature, which exploit simple traffic models (ON-OFF) for video traffic, could show unexpected behaviour if tested under a more realistic traffic model, due to the peculiarity of the MPEG traffic profile.

In this paper, we present realistic audio and video traffic models for simulation of MPEG sources based on ATM networks and propose interesting modifications to some MAC algorithms, objects of previous studies. The attention is on optimal radio resource allocation and QoS guarantees when using the multi-bearer transport and inner synchronisation mechanisms. This paper is organized as follows: Section II describes our MPEG-over-ATM traffic model; section III introduces performance indexes; section IV describes the proposed MAC algorithms; and section V reports simulation scenarios and performance results.

II. MPEG-OVER-ATM VIDEO AND AUDIO MODELS

According to the multi-bearer approach, video and audio streams of a MPEG flow can be transmitted separately over the radio interface. We describe the MPEG2-over-ATM models developed for our study to describe the generation of separate, i.e. not multiplexed, audio and video traffic streams.
A. Video traffic modelling

The MPEG-2 compression technique exploits video temporal and spatial redundancy to generate three types of photograms (frames): Intraframe (I), Forward Predicted frames (P), and Bidirectional frames (B). These frames are grouped together into a GOP (group of pictures), whose structure is identically repeated in the MPEG-2 stream (a GOP can be 15 frames, IBBBBBBBBBBBBBB). What we need is a statistical characterisation for these frames. The analysis of several MPEG2 clips [4, 5] showed that the size of I, B and P frames is log-normally distributed with different mean and variance values. From the pdf distribution, we can compute the frame sizes from a sample videoclip.

Each video frame is segmented into 184-byte blocks (with 4-byte header), that represent the MPEG transport streams (TS). If the generated frame length is not a multiple of 184 bytes, the last TS is suitably padded. In our model, we foresee the TS encapsulation into ATM adaptation layer (AAL) Service Data Units (SDU). We choose AAL-5, because it is recommended by the ATM Forum for MPEG video transport over CBR or VBR connections [1]. We assume that one AAL SDU contains all the TSs belonging to the same video frame. Every AAL5-SDU, with 8-byte header, is then fragmented into 48-byte cells; then 5-byte header is added to each cell to create the 53 bytes long ATM cell.

B. Audio traffic modelling

The reference MPEG audio model is the MPEG layer II and III. According to the specifications, it consists of a sequence of identical frames, each containing 1152 PCM 16-bit audio samples. The frame length for a stereo signal is given by [6]:

\[
\text{Frame Size} = 144 \times \frac{\text{Bit Rate}}{\text{Sample Rate} + \text{Padding}} \quad (1)
\]

If we assume 44.1 KHz sampling rate and 128 Kbps bit rate, each frame size is 417.96 bytes. The frame duration is related to the sampling rate and is 26.1ms. We assume that the multimedia terminal is equipped with silence-suppression technology [6], hence the resulting VBR audio traffic can be modelled with the one-way Brady model. It is an ON-OFF model with an average talkspurt duration of \( T_{on} = 1.41 \text{s} \), and an average silent period of \( T_{off} = 1.78 \text{s} \). Each audio frame is then fragmented into audio TSs with 4-byte header and 184-byte payload. Padding can be used in the last audio TS. TS packets are encapsulated in AAL SDUs, and the SDUs are fragmented and encapsulated into 53-byte ATM cells.

Audio with silence suppression based on a general-purpose two state model is described in literature. Numeric values for the state transition probabilities are not given. We realise that taking the values from the Brady model is an approximation not completely suitable for general-purpose audio. Nevertheless, for the purpose of testing inter-component synchronisation mechanisms the exact audio traffic profile description seems a minor concern. Thus, we are confident that the adoption of a more accurate model will confirm the results reported in the present paper.

In our models, we assume that the processing delay introduced by fragmentation and encapsulation is negligible compared to the transmission delay, but we obviously include the significant overhead in bits induced by fragmentation, encapsulation and control. This is not usually taken into account by simplified models, even if it strongly influences the algorithms performance, especially the delay and synchronisation degree among traffic flows.

III. AD-HOC METRICS

The performance is evaluated by means of ad-hoc metrics that measure the quality of audio and video streams of a real-time MPEG application. They are the end-to-end cell delay, the cell-loss ratio (CLR), and the maximum interarrival delay (jitter) between frames of the same stream. Furthermore, the maximum interarrival delay between cells of different multimedia streams of the same multimedia application determines the value of the inter-stream synchronisation index (\( \gamma \)).

To measure the synchronisation we need to “mark” the cells from media streams belonging to the same multimedia application. Furthermore, we need to choose a master component which acts as a reference for the maintenance of synchronisation. For an MPEG audio-video application, we select the video stream as the master service [7]. Every time an application generates a video frame, the same identifier will be assigned to all ATM video cells belonging to this frame. Audio cells generated during the same temporal interval will be marked with the same identifier. In other words, a logical input window (\( W_{in} \)) with the same duration of a video frame (33.36ms at 30 fps) is progressively numbered and associated to the generation of each video frame. Audio and video cells which are generated within \( W_{in} \) get the same identifier (ID).

Due to the delays caused by the random access protocol and the transmission scheduling process, a cell can be buffered for a certain amount of time before being transmitted over the radio channel. This is the reason for the introduction of an output window (\( W_{out} \)). Differently from \( W_{in} \), which is associated to the “generation” of a video frame, \( W_{out} \) can be associated to the “transmission” of either an audio (audio \( W_{out} \)) or a video (video \( W_{out} \)) frame over the air interface. Its duration is equal to the time necessary to transmit the whole (audio or video) frame over the radio interface. In other words, audio (or video) \( W_{out} \) starts with the beginning of transmission of the first audio (or video) cell with a certain ID and finishes with the end of transmission of the last audio (or video) cell with the same ID. It goes without saying that \( W_{in} \), video \( W_{out} \) and audio \( W_{out} \) identified with the same ID are shifted in time and can have different time durations.

The maximum tolerable end-to-end delay for a video source is in the range of 150-400ms [8], on the contrary audio has a lower delay tolerance, characterised by a maximum expiry time of 35ms. For the jitter measurement we assume, according to [9], that no smoothing technique is used to avoid the introduction of further processing delay. With no smoothing technique and a playback rate of 30 fps, the receiver will require a video frame about every 33ms. All the cells of the same video frame must be delivered within this temporal interval. Thus, each \( W_{in} \) must be completely delivered within 33ms, otherwise the frame is discarded.

To measure the jitter experienced by video/audio frames we use a clock, with a period equal to the \( W_{in} \) duration (i.e., one video frame duration). It starts as soon as the first cell of one (audio or video) component acquires the radio resource, i.e., the multimedia transmission begins. This clock defines a sequence of temporal intervals (clock windows), each with
the duration of an input window. Clock windows are numbered with a progressive ID and they logically represent the expected sequence of transmissions of the $W_{in}$ windows.

In so doing, the jitter for each component stream can be expressed as the difference between the end time of the (audio or video) $W_{out}$ with a given ID (computed with reference to the start of the clock) and the end time of the associated clock window (with the same ID).

Figure 1 illustrates the delay jitter computation for a video stream.

$$\text{delay jitter} = (T_2 - C_1)$$  \hspace{1cm} (2)

Fig. 1. Delay Jitter between video frames of the same stream.

The same considerations apply to the audio source; the corresponding frame period (with 44.1KHz audio samples and a 128 Kbit/sec) is equal to about 25 ms.

By referring to the sequence of clock windows we can also evaluate the degree of inter-stream synchronisation. The synchronization index $\gamma$ gives a measure of the relative delay between cells of different traffic streams belonging to the same multimedia connection. An excessive delay will result in completely desynchronised audio and video streams. We can evaluate the desynchronisation degree by measuring the delay jitter between video $W_{out}$ and audio $W_{out}$ with the same ID, as illustrated in Figure 2. The jitter gives a measure of the inter-stream synchronization: the higher the delay the greater the desynchronisation degree. When the delay jitter is higher than 80ms [9], the frames are completely discarded.

$$\gamma = 1 - \frac{\text{delay jitter}_{AV}}{\text{Rit}_{MAX}} = \begin{cases} 1 & \text{for } \text{delay jitter}_{AV} = 0 \\ 0 & \text{for } \text{delay jitter}_{AV} \geq 80 \end{cases}$$  \hspace{1cm} (4)

where Rit_{MAX} is equal to 80ms.

In [10] we defined the following QoS metric for each monomedia component:

$$QoS_{\text{partial}} = \frac{1}{2} \cdot \frac{\text{CLR}_{\text{partial}} \cdot \text{CLR}_{\text{MAX}}}{\text{CLR}_{\text{MAX}}} + \frac{1}{2} \cdot \frac{\text{wt}_{\text{MAX}} - \text{wt}_{\text{partial}}}{\text{wt}_{\text{MAX}}}$$  \hspace{1cm} (5)

The global QoS of the whole multimedia application is given by the normalized sum of the partial QoSs of the monomedia components. For an audio-video application we have:

$$QoS_{\text{audio-video}} = 0.5 \cdot QoS_{\text{video partial}} + 0.5 \cdot QoS_{\text{audio partial}}$$  \hspace{1cm} (6)

IV. PROPOSED MAC ALGORITHMS

The algorithms described in this paper are enhancements of the one we presented in [10], called Urgency-Based Dynamic Reservation (UBDR) protocol. For the sake of clearness, we briefly report the main features of the UBDR protocol. It is TDMA/TDD based and uses a centralized architecture with the Base Station (BS) able to control cell transmissions over the radio interface. The UBDR frame has fixed length and consists of variable-length subframe fields: frame header, contention field, uplink, and downlink fields.

Mobile terminals ask for bandwidth reservation by using contention slots in random access. The transmission scheduling policy is mainly based on the urgency parameter $P$. This parameter is computed directly by the mobile terminal on the basis of its internal status (i.e. urgency of the cells queued in the buffers). At the end of the $i$-th frame, the scheduler in the BS receives the $P$ values from the whole set of active connections; it computes the corresponding amount of resources that will be allocated to each of them during frame $i+1$, and broadcasts the results to the terminals. Each active terminal transmits its buffered cells by using the bandwidth resources assigned by the BS for the $(i+1)$-th frame. Once transmitted, if its buffers are empty, a terminal enters the sleeping mode, otherwise the cycle is repeated.

The definition of $P$ we consider in this paper takes into account the elapsed waiting time of the queued cells:

$$P_i = \text{wt}/\text{wt}_{\text{MAX}}$$  \hspace{1cm} (7)

where $\text{wt}$ is the delay accumulated by the first cell queued in the buffer at the end of frame $i$ and $\text{wt}_{\text{MAX}}$ is the maximum allowed cell waiting time before the dropping occurs.

The BS calculates the number of slots assigned to connection $j$ as follows:

$$X_{ij} = \min \{\text{num\_slot}_{i-1,j}, \sum_{k=1}^{Z} P_{ij}^{k} \cdot N_{i} \}$$  \hspace{1cm} (8)

where $\text{num\_slot}_{i-1,j}$ is the number of slots necessary to transmit all the cells remained in the buffer of terminal $j$ at the end of frame $i-1$, $N_{i}$ is the number of slots available for transmission, $P$ is the urgency parameter, and $Z$ is the number of active connections in frame $i$ (please refer to [10] for details).

In this paper, we enhance the UBDR algorithm to meet the QoS and synchronisation exigencies of multimedia terminals and applications. According to our view of multimedia terminals and to the multi-bearer approach, each monomedia in the terminal has its own buffer. It is independent from the others, as each monomedia flow receives a given
amount of slots, according to its own urgency parameter \( P \). Early studies and analysis, not included in this paper, showed that the control overhead introduced by the use of multibearer links can be easily reduced to negligible values.

A. CMS Algorithm: UBDR with Control mini-slots

A feature which has a deep impact on the reservation phase of the UBDR algorithm is the use of control mini-slots within the reservation sub-frame. When a connection is active (its buffer is not empty), it can use the piggyback to communicate the value of its parameter \( P \). When the connection is passing into the silence state, the last cell in the buffer is used to inform the BS, which allocates a control mini-slot (4 bytes) to the silent connection. As soon as a new cell arrives at the transmission buffer of the connection, the terminal can send information about its buffer status to the BS through the assigned mini-slot. This mechanism provides a contention-free access for the accepted connections, which can transmit in the frame immediately following their activation. We have preferred this approach due to the statistical nature of the MPEG traffic streams, which tends to empty the buffers frame-by-frame. This approach of course could cause a slight bandwidth waste, because the BS must reserve as many mini-slots as the number of monomedia components originating from the multimedia terminal. Notwithstanding, this wasting is limited by practical considerations: the W-ATM system works in a typical picocellular environment [11], so the number of terminals is somewhat bounded by the dimension of the cell itself and cannot grow infinitely.

B. SYN Algorithm: CMS with clock- synchronised transmission

This algorithm aims to improve the synchronization between monomedia flows by operating on the CMS scheduling. The idea is to put into relation the different flows belonging to the same connection by means of the sequence of temporal intervals determined by the clock windows. As pointed out, these windows tell us which should be the expected temporal sequence and duration over the air interface of the input windows generated by the sources with a given ID. Through the sequence of clock windows the degree of synchronization of the components can be monitored and maintained.

The algorithm modifies the slot-assignment phase operated by the BS. In particular, if we define with \( Out_{ID} \) the current clock window identifier, we have:

\[
X' = \min \left\{ num_{slot}^{i,j}_{i-ID}, \sum_{P_j}^{P_i} N_i \text{ for } ID \leq Out_{ID} \right\}
\]

where \( num_{slot}^{i,j}_{i-ID} \) is the number of slots necessary to transmit all the cells with an ID inferior or equal to \( Out_{ID} \) in the buffer of the terminal \( j \). In fact, these cells are the delayed video or audio cells and must be transmitted; the other cells with ID higher than \( Out_{ID} \) remain in the buffer, as they are anticipated (early) cells.

V. SIMULATION RESULTS

A simulation campaign has been performed to assess the effectiveness of the proposed MAC algorithms. The system is loaded by multimedia terminals. Each of them generates MPEG video and audio flows according to the described models. The MPEG video and audio parameters are the same for all multimedia terminals and are extracted from a sample MPEG stream [5]. They are reported in Table I.

All the curves reported in this section are sketched versus the normalised system load. The maximum load of 1 corresponds to the maximum number of 34 terminals, i.e. 68 connections.

<table>
<thead>
<tr>
<th>Parameters for performance evaluation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. MAIN SYSTEM PARAMETERS</td>
</tr>
<tr>
<td>Total channel capacity</td>
</tr>
<tr>
<td>Frame duration</td>
</tr>
<tr>
<td>Slot size</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D. MPEG Video source parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Frame mean size [Kbit] / std deviation</td>
</tr>
<tr>
<td>B-Frame mean size [Kbit] / std deviation</td>
</tr>
<tr>
<td>P-Frame mean size [Kbit] / std deviation</td>
</tr>
<tr>
<td>Frames rate (fps)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E. MPEG audio source parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak Rate</td>
</tr>
<tr>
<td>Sample Rate</td>
</tr>
<tr>
<td>Mean ON period duration (Ton)</td>
</tr>
<tr>
<td>Mean OFF period duration (Toff)</td>
</tr>
</tbody>
</table>

The utilization efficiency, in terms of the percentage of uplink bandwidth actually exploited by the terminals has been evaluated for the UBDR, CMS, and SYN algorithms. Curves are not shown for paper length constraints, anyway all algorithms show an almost identical behaviour.

In Figures 3 and 4 the mean delay jitter for video and audio components is reported. The UBDR algorithm shows the worst jitter, while it remains negligible for both CMS and SYN algorithms even when the system load increases.

Fig. 3. Mean Video Delay Jitter.

If we consider the values of the \( \gamma \) parameter, which takes into account audio-video synchronization, in Figure 5 the worst behaviour is obtained by the UBDR algorithm. This bad performance is mainly caused by the lack of a synchronization technique and by the particular statistical nature of MPEG traffic streams which frequently empties audio and video buffers. The connections are then forced to ask for reservation through the random-access slots. This
leads to potential collisions and, consequently, cells can be delayed (or discarded) until reservation occurs. CMS, thanks to its collision-free access, shows very good performance. SYN, as expected, achieves the best results under every load.

We have also investigated the impact on QoS performance. Figure 6 shows the average QoS for the whole application generated by a multimedia terminal. The best performance is achieved by the SYN algorithm followed by CMS. UBDR shows the worst performance, due to the great number of collisions which delay the access and lengthen the waiting time of the cells in the buffers. The audio and video cell loss rate (CLR) increase, this adversely affecting the overall QoS. Also the average QoS for video and audio monomedia traffic components is measured (curves not shown), and results are analogous.

It is worth highlighting that the audio stream is particularly affected by the SYN protocol. It shows a superior performance, with respect to CMS and UBDR, thanks to its capability of forcing the contemporary transmission of video and audio cells with the same ID. The only drawback is represented by an increased mean waiting time of audio cells in the buffers. This particularly affects the QoS of audio streams in SYN when compared to CMS. Nevertheless, the CLR of audio streams remains low and the overall QoS performance in not affected, thanks also to the high QoS of video flows.

VI. CONCLUSIONS

In this paper we investigated the performance of novel techniques for the transport of multimedia applications over a Wireless ATM architecture. In particular, we focused our attention on the impact that applications generating MPEG-coded traffic flows have on the performance of different MAC protocols. We exploited a non-integrated MPEG over ATM model for video and audio traffic and proposed some features to overcome QoS and synchronization impairments experienced by the multimedia traffic transported across separate bearers. Achieved results seem very encouraging, and worth being extended to systems different from W-ATM.

REFERENCES