Multimedia Tools and Applications, 25, 187–216, 2005

AVP: A Highly Efficient Transport Protocol for Low Bit Rate Multimedia Communications

JIANYU DONG jdong2@calstatela.edu
CHAO HE hec@ee.eng.ohio-state.edu
YUAN F. ZHENG zheng@ee.eng.ohio-state.edu
Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210, USA

ROBERT L. EWING
Information Directorate, AFRL/IFTA, Wright Patterson AFB, OH 45433, USA

Abstract. Utilization of Internet communications in distance learning, distributed simulation, and distributed work groups involves multimedia transmission of animation, voice and video clips. Highly compressed audio-video data protocols are required for efficient Internet multimedia communications. Addressing this requirement, a new transport protocol called Audio-Video Protocol (AVP) for highly efficient multimedia communications on the Internet is presented. While providing similar real-time delivery functions as Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP), AVP adopts a novel audio-based synchronization scheme. This synchronization scheme has two advantages. One is the overhead reduction through eliminating the timestamp in each transmitted data packet. The other is the packet rate reduction by putting multiple audio frames or mixed audio-video frames in a single AVP packet. As a result, the end-to-end media unit delay is reduced while achieving implicit synchronization. Furthermore, AVP provides adaptive quality of service (QoS) by the prioritized packetization scheme. Simulation results are presented to verify the advantages of the AVP protocol.

Keywords: RTP/RTCP, video streaming, synchronization, end-to-end delay, QoS

1. Introduction

Multimedia applications on the Internet have become more and more popular in recent years. Video streaming is specially attractive because people can enjoy the video online without downloading files of big size. The amount of multimedia content available in digital archives, on the World Wide Web, in broadcast data streams and in medical and technology databases is growing rapidly. This large context of multimedia data has led to increasing difficulties in accessing, identifying and displaying such resources due to their volume and complexity. A well-designed transport protocol, with the ability to provide both timely delivery and good synchronization for multimedia data, will be extremely valuable. TCP and UDP are two common-used transport protocols in current Internet. TCP is not suitable for video streaming since retransmission will cause a heavy traffic burden. Compared to TCP, UDP can provide prompt delivery, but its structure is too simple to handle the packet loss, duplication, and reordering by itself. To improve the streaming performance, Real-time Transport Protocol (RTP) [23] is often used on top of UDP for synchronization and QoS control [1]. As a well-accepted real-time protocol, RTP is widely used not only in

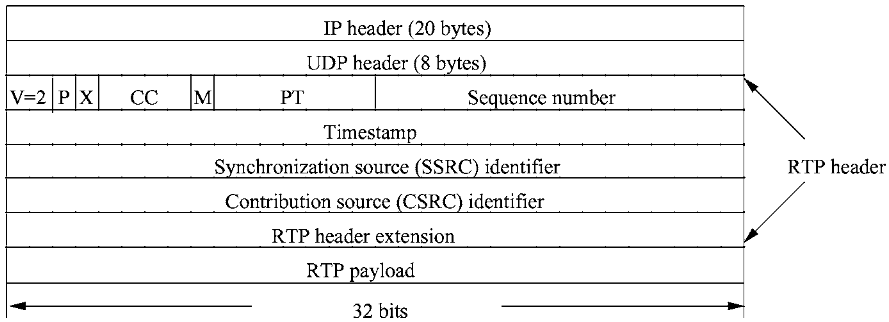

Figure 1. Packet header for RTP payload.

the interactive applications such as videoconference [7, 9], but also for delivery of stored multimedia such as video streaming [13, 20, 21]. However, since RTP is originally designed for interactive applications and there exists significant difference between interactive and non-interactive applications, the efficiency of using RTP in the latter is questionable. First, the size of the RTP header is quite large. As shown in figure 1, the standard size of the RTP header is 96 bits (not including the CSRC field which is optional). Adding the 20-byte IP header and the 4-byte UDP header, the bandwidth needed to transmit the packet header for an RTP payload is 5.76 kbps if the packet rate is 20 per second, which takes a big share of the bandwidth for a 28.8 kbps modem. Apparently, the header size of RTP is an ineligible disadvantage of narrow-band applications. Second, RTP is intended for transmitting different media via separate sessions, which is desirable in videoconference to meet different QoS requirements. One disadvantage of the multisession delivery is the lack of efficiency due to the large overhead. Since audio and video are already in-sync at the server, and are always transmitted together in streaming applications, it is possible to design a single-session scheme which is able to provide both good transmission efficiency and adaptive QoS to different media.

In fact, the inefficiency of RTP has been noticed by many researchers. To reduce the overhead, an RTP header compression method was developed [2]. This compression method is based on the assumption that in some fields of the RTP header such as sequence number and timestamp, the second-order difference is zero. This assumption is valid for applications such as voice over IP where the increment of the timestamps of adjacent packets is a constant. However, the situation is different in video streaming. The constant assumption does not hold if the video is compressed by MPEG2 or some frame variable codecs where the timestamp will not increase monotonously or the increment will vary in a large range. Since these kinds of video codecs are often used, the RTP header can not be efficiently compressed. In addition, the compressed header can not handle packet loss very well. Experiment results show that packet loss rate ranged from 2% to 10% is common on the Internet. Therefore, robustness should always be a design consideration, especially in low bit rate applications.

In order to improve the efficiency of multisession transmission, a payload format (BMPEG) was designed for RTP to improve the transmission efficiency of streaming

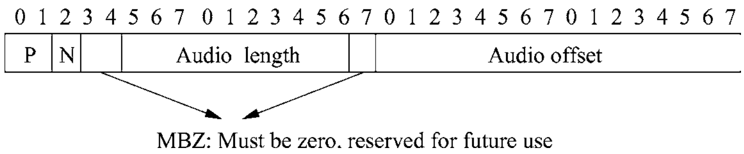

Figure 2. BMPEG specific header.

applications for MPEG video [3]. The basic idea for this payload format is to bundle MPEG video and audio together in one RTP packet to get rid of small audio packets. The BMPEG payload type can reduce the packet rate while achieving implicit synchronization. However, the efficiency improvement is quite limited. Since RTP uses timestamps for synchronization, two timestamps are needed in one RTP packet for BMPEG, one for video and the other for audio. Hence, the BMPEG packet carries an extensive header of 32 bits in addition to the 96-bit RTP header as shown in figure 2. The field *Audio Offset* is the skew between the video timestamp and the audio timestamp in the same packet so that the receiver can recover the audio timestamp for playback. As a result, the size of the header is actually bigger in this case and the overhead reduction only comes from the reduction of the total packet number. Furthermore, there are a few restrictions to the BMPEG payload. First, it can not monitor and control the QoS of video and audio separately. Secondly, it is not applicable for low frame rate video transmission [3] since the filed *Audio Offset* in the BMPEG extensive header is only 16 bits and the skew which it can represent is limited when the audio sampling rate is high. Finally, this payload format is only suitable for the media compressed by MPEG, not feasible for other compression methods such as 3-D wavelet compression [14, 16, 26]. In the 3-D wavelet compression, a group of video frames is accumulated and 3-D wavelet transform is applied to the whole group, and the result is arranged by the subbands in the frequency domain other than the frames in the time domain [26]. As a result, the video packets for the whole group have to carry the same timestamp in RTP. In this case, it is very difficult to bundle the audio with the video packets since a group of video could span a long time and it is impossible to record the skew between video and audio using the method in BMPEG. Because of the above restrictions, BMPEG has not been widely accepted for Internet applications where QoS should be monitored, and where low video-frame rate is frequently encountered.

As stated above, both the RTP header compression method and the BMPEG payload type have their own disadvantages, and the improvement of efficiency is limited. Currently, many home network users still use dial-up connection. For them, efficient usage of bandwidth is critical. In order to improve the efficiency and quality of video streaming on the Internet, a new protocol should be designed. In this paper, we propose a new Audio-Video Protocol (AVP) which provides good QoS to both unicast and single-channel multicast streaming applications with improved efficiency. The major feature of AVP is a new synchronization scheme which uses audio as the time base for different media. This is possible because the sampling rate of the audio signal contains the time information itself. As a result, embedding audio with video gives video the time reference and no timestamp is needed. With a smaller header and reduced packet rate, AVP can improve transmission efficiency better than BMPEG. AVP also adopts a prioritized packetization scheme to provide different

QoS for packets with different priorities. Thus, AVP has the advantages of BMPEG but has no restrictions as cited earlier.

This paper is organized as the following. In Section 2, we introduce the features of AVP. The details of the AVP structure are described in Section 3. Section 4 analyzes the performance of AVP. Simulation results are presented in Section 5. Section 6 summarizes the paper.

2. Features of AVP

2.1. The audio-based synchronization scheme

Synchronization is a special QoS requirement for multimedia applications. There are two kinds of synchronization requirements in video streaming, namely intra-media and inter-media. Intra-media synchronization is the continuousness of one media stream; inter-media synchronization often refers to the lip-synchronization. In order to achieve good synchronization in media play, the transport protocol should provide sufficient timing information for appropriate scheduling. Timestamping is a popular method for doing so. The role of timestamping is to record the time information and attach it to the data in a pre-defined format. For example, RTP/RTCP uses RTP and NTP (Network Time Protocol) timestamp [19] to provide the relative and absolute time information for the transmitted data, respectively. The advantage of timestamping is in its generality. That is, it is independent of the characteristics of the transmitted media. The shortcoming is that each packet should carry a timestamp as an overhead. The audio-based synchronization scheme which we propose can provide the required time information while eliminating the timestamp. The basic idea of the audio-based synchronization scheme is to embed video frames with their associated audio. It is different from the idea of the bundled MPEG, which simply combines video and audio frames together but still relies on timestamps for synchronization [3].

To achieve the audio-based synchronization, the data must be packetized properly. At the sender's side, a synchronized media group is formed by combining audio with its associated video, and an identical group number is assigned to each member packet of the group. There are three requirements in forming a synchronized media group. First, the audio/video within the group must be self-presentable, i.e., the audio/video can be presented right away when all the data packets of the group are received and reconstructed. Secondly, the first video frame in the group should be packed with the closest audio frame and the span of the audio frames should be as close as possible to the span of the corresponding video frames. The audio frame here is a segment of the audio signal of a fixed length. The length should be the same as defined in the corresponding audio compression method. Note that the time interval between two video frames is not necessarily a multiple of the length of the audio frame. Thirdly, the group can have more than one video frame, but should be small enough to meet the delay requirement of a specific application. At the receiver side, the first video and audio frames are played at the same time, and the following video frames are displayed according to the video frame rate (information such as video frame and audio sampling rates should be transmitted before the start of data transmission using a reliable

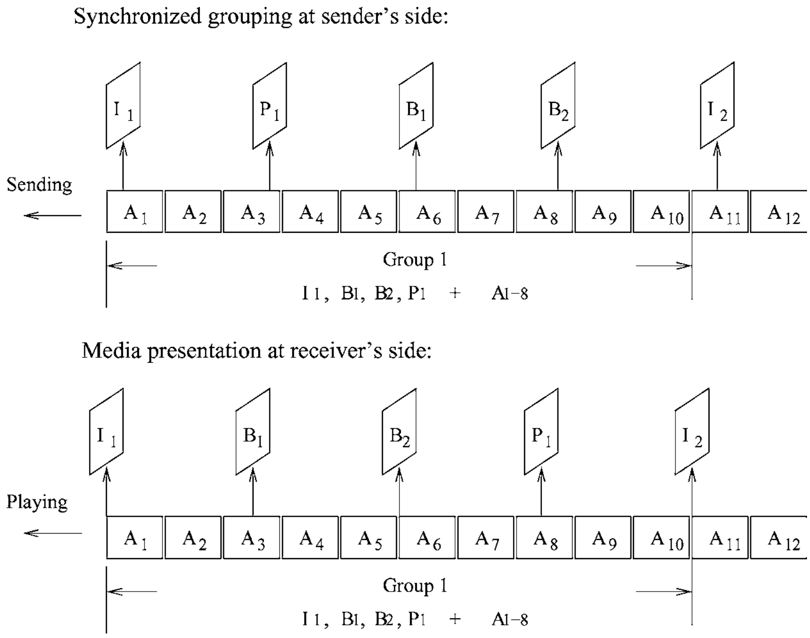

Figure 3. Audio-based synchronization for MPEG-2 video.

protocol). To describe the grouping procedure more clearly, we use the MPEG-2 video as an example.

In MPEG-2, the frame structure of video bitstream is I-B-B-P..., the order of the presentation is still I-B-B-P but the order of transmission is I-P-B-B because the B frames are compressed and sent after the corresponding I and P frames. In this case, the P frame can be reconstructed upon its arrival, but can not be displayed until the two B frames are reconstructed and displayed. This kind of P frames is not self-presentable. As a result, the P frame must be grouped with the following two B frames to form a synchronized group. Figure 3 shows the grouping procedure.

From figure 3, one can see that this audio-based synchronization scheme may generate a skew if the audio is compressed by a frame-based algorithm. However, the skew should be less than one half of the audio frame length since the video is always grouped with the closest audio. According to the research on human perception [22], the video and audio can be considered in-sync if the skew is less than 80 ms, and a skew within the range [−160 ms, 160 ms] is quite acceptable. As the length of an audio frame seldom goes beyond 35 ms in the popular audio compression algorithms as shown in Table 1, the skew caused by grouping can be ignored.

Hence, the audio-based synchronization scheme is in-sync enough for human perception. Also a good property of this scheme is that synchronization is achieved for each group. Even when synchronization is lost for one group, it is automatically recovered at the beginning of the next group.

Table 1. Frame lengths in audio compression algorithms and maximum skews by AVP.

	Sampling rate (kHz)	Samples per frame	Frame length (ms)	Max skew (ms)
MPEG Layer 1	44.1	384	8.71	4.35
MPEG Layer 2	44.1	1152	26.1	13.05
G.722	16	320	20	10
G.728	8	20	2.5	1.25
GSM	16	320	20	10

2.2. *The prioritized packetization scheme*

Since the audio-based synchronization is implemented based on groups, the AVP packetization can be very flexible. Video and audio can be put together in one packet or sent separately according to the QoS requirement and network conditions. There are three types of packets in AVP: audio only packet, video only packet and mixed packet. The definitions are shown below.

- *Audio only packet*: Packet that contains only the audio signal. One or more audio frames may be presented in an audio only packet.
- *Video only packet*: Packet that contains only the video signal.
- *Mixed packet*: Packet that contains both audio and video signals.

The flexibility of the AVP packetization makes it possible to provide different QoS' to video and audio. For example, when the bandwidth is enough, the sender will packetize audio and video together to generate the mixed packets; when congestion happens, the sender may drop the video frames and send the audio only packet, as shown in figure 4.

Dynamical QoS control is very important to multimedia applications. QoS control at the application level can be viewed as the end-to-end control because this kind of control only affects the end users. If different QoS' for different users are needed, the control method should be implemented at lower levels. To reach this purpose, a prioritized packetization scheme is adopted which assigns different priorities to the AVP packets according to their contents. Packets containing audio has the highest priority 00. For video-only packets, different priority numbers are assigned according to the importance of the content in the packet. For example, the low band coefficients have a higher priority than the higher band coefficients, if the wavelet compression is used, because the former contains most energy of the original signal [16, 24]. If the MPEG compression is used, the codes of the I frame have a higher priority than that of the other frames since it serves as a reference for predicting the other frames [12].

With the prioritized packetization scheme, the intermediate systems such as the translators at the transport level can offer different QoS' to different packets. The scheme can be even better supported with the implementation of IPv6 in the near future. In the IPv6 header, a field called *Traffic Class* is used to distinguish different classes of packets [11]. Taking this advantage, we can label the priorities of AVP packets in both AVP and IP headers as

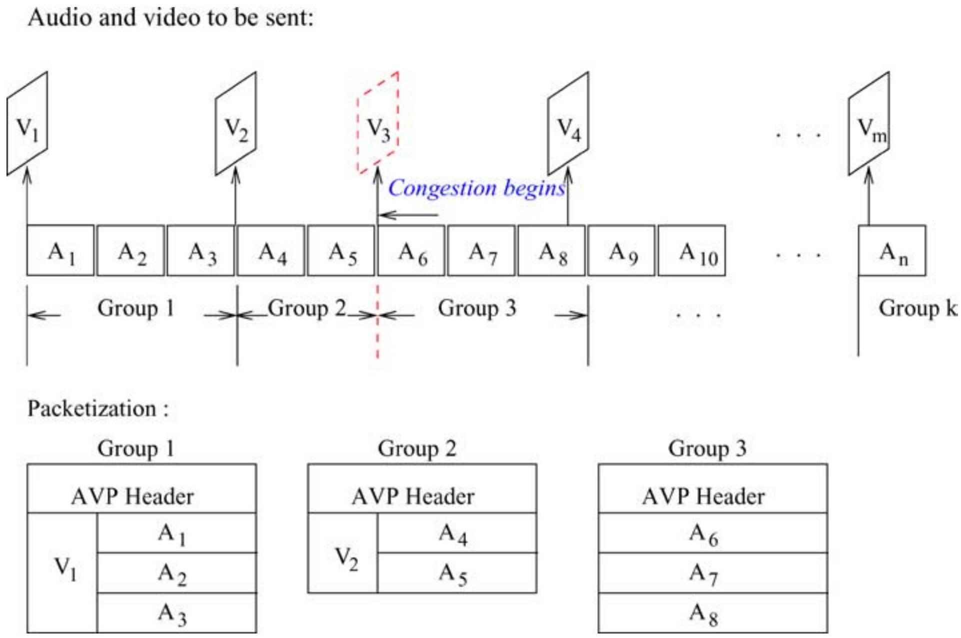

Figure 4. AVP packetization changes with network condition (V_3 in Group 3 is dropped when the network congestion begins).

shown in figure 5. Thus dynamical QoS control can be achieved not only in the transport layer but also in the network layer. For example, in the transport layer, the translator can only forward the packets with the highest priority to some addresses where the bandwidth is limited; in the network layer, the packets with lower priorities can be dropped first at routers when congestion occurs.

Furthermore, the prioritized packetization scheme enables AVP to achieve prioritized selective retransmission. Although retransmission is not desirable in multimedia applications since it might add a heavy burden to the traffic, some applications may choose this function to ensure the transmission of high priority packets; therefore, AVP keeps it as an option.

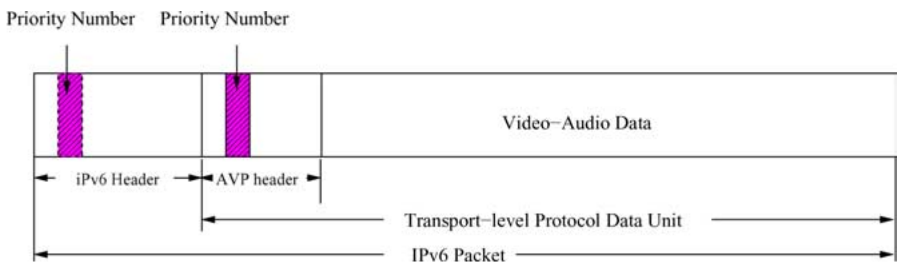

Figure 5. Structure of IPv6 packet with prioritized packetization scheme.

The details of the prioritized selective retransmission scheme will be introduced in the next section.

3. The audio-video transport protocol

As described above, AVP has new synchronization and packetization schemes to address the special needs of video streaming. Besides these new features, it overall structure is similar to RTP/RTCP, i.e., it consists of two parts: data transport protocol and control protocol. The data transport protocol (AVP) works with the underlying protocols to provide multimedia data transmission, while the control (AVCP) protocol provides means for controlling the QoS.

3.1. The structure of the AVP header

As a transport protocol for multimedia communication, AVP is able to provide synchronization information in addition to other functions such as packet reordering and loss detection. Furthermore, AVP can deal with packet loss and long-delay to provide a satisfactory performance. The fields in the AVP fixed or extensive header are designed to achieve the above goals.

3.1.1. The AVP fixed header. The structure of the AVP fixed header is shown in figure 6, and the definition of each field in the header is listed below:

- *Priority number* (*PN*): 2 bits. This field identifies the priority of the packet according to the priority scheme.
- *Extension* (*X*): 1 bit. When it is set to 1, the AVP header will be followed by an extensive header.
- *Mark* (*M*): 1 bit. It indicates the boundary of the AVP packet stream; usually the bit is set in the last packet of a group.
- *Multiple source indicator* (*I*): 1 bit. It indicates if there are multiple sources presented in this application.
- *CSRC count* (*CC*): 3 bits. It indicates the number of the contribution source identifiers (CSRC) that follows the fixed header. The definition of this field is the same as that in RTP [7].

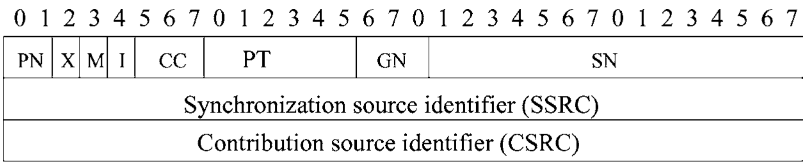

Figure 6. AVP fixed header fields.

- *Payload type* (*PT*): 6 bits. This field defines the format of the payload. Typically, it indicates a specific combination of audio and video, such as MPEG2 video with MP3 audio, or 3D wavelet video with G.729 audio. AVP has a standard table to index frequently used combinations. If all the six bits are set to 1, it is a user-defined payload type.
- *Group number* (*GN*): 3 bits. An identical group number is assigned to all the packets of a synchronized group. Since the group size is usually quite large, 3-bit is sufficient for general streaming applications.
- *Sequence number* (*SN*): 15 bits. This field can be used for receiver to rearrange received data packets and detect packet loss. This definition is the same as that in RTP.
- *SSRC*: 32 bits. It is valid if **I** is set to 1. This field identifies the synchronization source similar to that defined in RTP [7].
- *CSRC*: 32 bits each. The field is presented when CC is not zero. This field identifies the contribution source as defined in RTP [7].

When AVP packs video and audio from the same source into one single stream, the fields *SSRC* and *CSRC* are not necessary in single-source applications; therefore, the two fields are optional in AVP. This design lets AVP work more efficiently in single-source applications and reserves the ability to handle applications with multiple sources such as videoconferencing.

3.1.2. The AVP extensive header. In addition to the fixed header, an AVP packet might carry the extensive header. The structure of the extensive header is shown in figure 7. The function of the extensive header is to provide redundant information for better error recovery and concealment when packet loss occurs. Since packet loss may affect the synchronization, the extensive header is necessary for some payload types, e.g. variable bit rate (VBR) audio. The following extensive headers have been designed for different codecs:

- Constant bit rate (CBR) audio codecs:
  This is a simple case. For the packets containing audio (with priority number 00), only one field is added to indicate the number of packets with priority 00 in the current synchronized group. The structure is shown in figure 8.
- Variable bit rate (VBR) audio codecs:

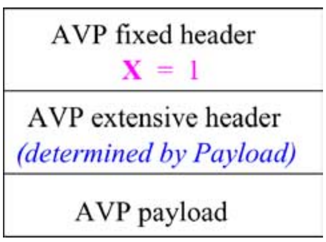

Figure 7. The structure of the AVP extensive header.

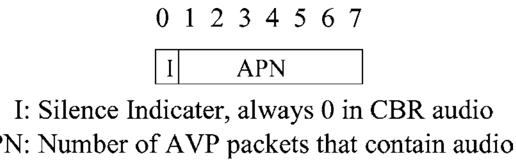

Figure 8. The AVP extensive header for CBR audio.

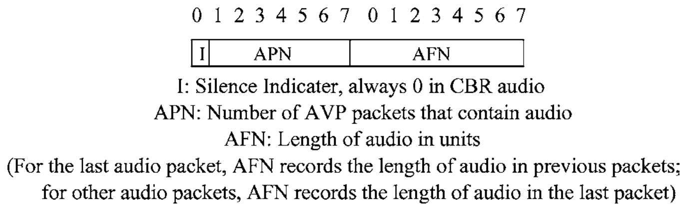

Figure 9. The AVP extensive header for VBR audio.

This case is more complicated than the one shown above because silence suppression is enabled in the variable bit rate audio codec. That is, the audio codec can detect and discard silent audio frames such that only "loud" audio frames are encoded and transmitted. In this case, the AVP extensive header uses *Silence Indicator* to indicate if there is silence in this packet. If there are one or more silence periods, the format of the payload will be changed. One bit will be inserted to indicate if the following data is silence or not. In addition, more redundant information is embedded such that the receiver can achieve synchronization in case of packet loss as shown in figure 9. Hence the total length of the extensive header is increased to 2 bytes in this case.

- Frame variable (FV) video:

  When the rate control is used in the video codec, some video frames may be skipped. To deal with it, a 4-bit video extensive header is attached to all the packets that contain the video signal. The function of the video extensive header is to record the play position of the video frame. For example, if no frame is skipped, the play-position of the the first frame in a synchronized group is 1; if two frames are skipped before the first frame, the play-position of the first frame is 3. When the receiver plays the video according to the frame rate and the play-position, the time information between the video frames can be exactly recovered. For some long-delay applications, the length of the video extensive header can be extended accordingly. Also the video extensive header can be helpful in framing. If some framing information in the payload is lost because of the packet loss, the receiver can use the video extensive header and the group number for framing. That is, the packets with the same play position in a group belong to the same frame.

3.2. *AVP packetization*

A good transport protocol should perform fairly well even when packet loss, packet duplication, or long-delay occurs [10]. Like RTP, the AVP packetization scheme is also consistent with the concept of Application Level Framing (ALF) [4], which means that the receiver should be able to process an Application Data Unit (ADU) delivered by a transport protocol such as a video frame even if the previous ADU is lost. In order to be more robust to packet loss, the AVP packetization obeys the following rules:

- The size of an AVP packet should be smaller than the size of MTU (Maximum Transmission Unit) of the network. This is to avoid further fragmentation at lower levels.
- An AVP packet is self-decodable. There should only be complete audio frames in one packet. Partial video frame is allowed in one packet, but additional information should be included in the payload format to make it decodable.
- The length of audio in each AVP packet that contains audio should be the same except in the last packet. Since we use the number of audio frames to indicate the length of audio, the number of audio frames allowed in each AVP packet should be the same. If silence is presented, the sum of the silence length and the audio length should be the same in each packet except the last one. The length should be indicated in the audio extensive header. Figure 10 shows how to packetize the audio frames in AVP to support silence suppression, in which a shaded square represents a silence frame.

With the above packetization rules, the effect of packet loss can be minimized. The following sections shows how to packetize video and audio data for a few compression methods.

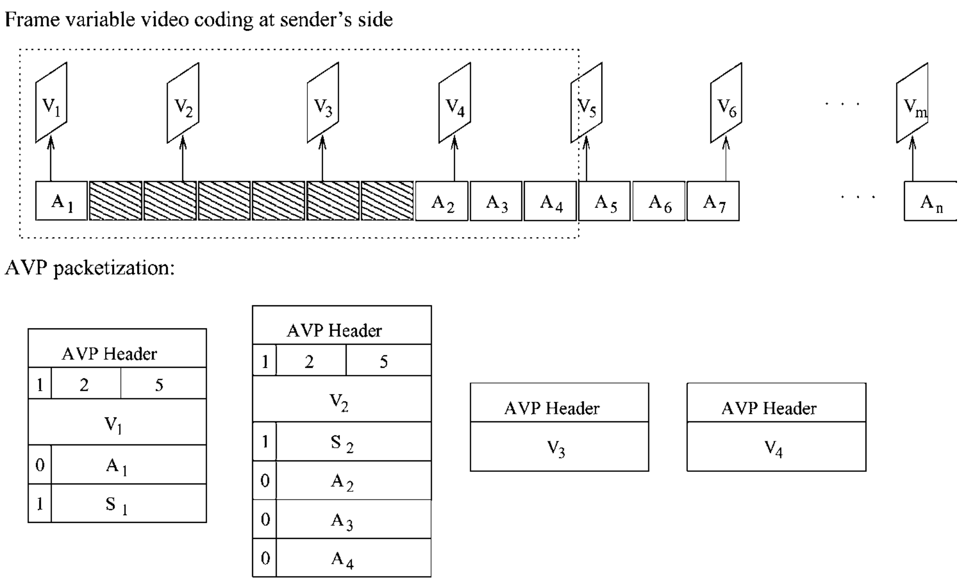

Figure 10. AVP packetization for VBR audio.

3.2.1. AVP packetization for 3-D wavelet video. It is easy to implement the AVP packetization in the 3-D wavelet-based compression. Since in the latter, a group of video frames is collected to perform the 3-D wavelet transform, AVP can simply use the video frames and the associated audio to form the synchronized group. In packetization, AVP can combine multiple audio frames with the Low-Low (LL) band wavelet coefficients to form the packets of priority 00. The LL band is the most fundamental band in the multi-band analysis of a video signal in the wavelet transform [17]. With the wavelet coefficients of the LL band, a video signal of low resolution can be recovered even without the coefficients of the other bands. This is the reason why a high priority number is assigned to the packets containing the LL band data.

3.2.2. AVP packetization for frame variable video. In the audio-based synchronization scheme, the video frames are played according to the pre-known frame rate. However, frame variable coding is widely used in low bit rate communications. For example, in the H.263+ Test Model 8 [8, 15], one or more video frames are skipped according to the available buffer size. Although the frame variable codec skips frames here and there, the frame sampling rate is fixed; therefore, the basic synchronization scheme of AVP is still valid. Since the position of the video frame is indicated in the video extensive header, the receiver can simply play back the video frame according to the position number (field *VPS*). In this way, the frames skipped by the encoder will also be skipped by the player, as shown in figure 11.

3.2.3. AVP packetization for variable bit rate audio. In AVP, silence suppression is supported by the audio extensive header and the packetization rules. Generally, for audio-video delivery, if silence presents at the end of a synchronized group, we need not pack it into the AVP packet since the time information can be recovered by the video frame rate; otherwise, the silence information should be included in the AVP packet as shown in figure 10. In the AVP audio extensive header, the length of the silence period is represented by the number of the audio frames which it covers. For example, in figure 10, the first AVP packet contains one audio frame A_1 and one silence period S_1. Since the length of S_1 is four times the length of a audio frame, its length is 4, and the total length of audio is $1 + 4 = 5$ which is the value of the field *AFL* in the AVP extensive header for the VBR audio.

3.3. AVP control protocol (AVCP)

Similar to RTCP, AVCP also has the sender's report (SR) and the receiver's report (RR). Since the inter-media synchronization is achieved via the audio-based scheme, there is no need to include relative time information such as the RTP timestamp in the sender's report. However, the NTP timestamp in RTCP is preserved. Since the SR contains both the sequence number and the NTP timestamp of a packet, the receiver can know its absolute sending time. At the beginning of the media delivery, a AVCP packet will be sent with the first data packet to indicate its sending time. Assuming that the sending time of the first packet is T_1 and the time span of the first synchronized group is T_{g1}, the receiver can get the sending time of the next group which is $T_1 + T_{g1}$. This information is helpful in scheduling the proper display time of a synchronized group. Also the NTP timestamp can be combined

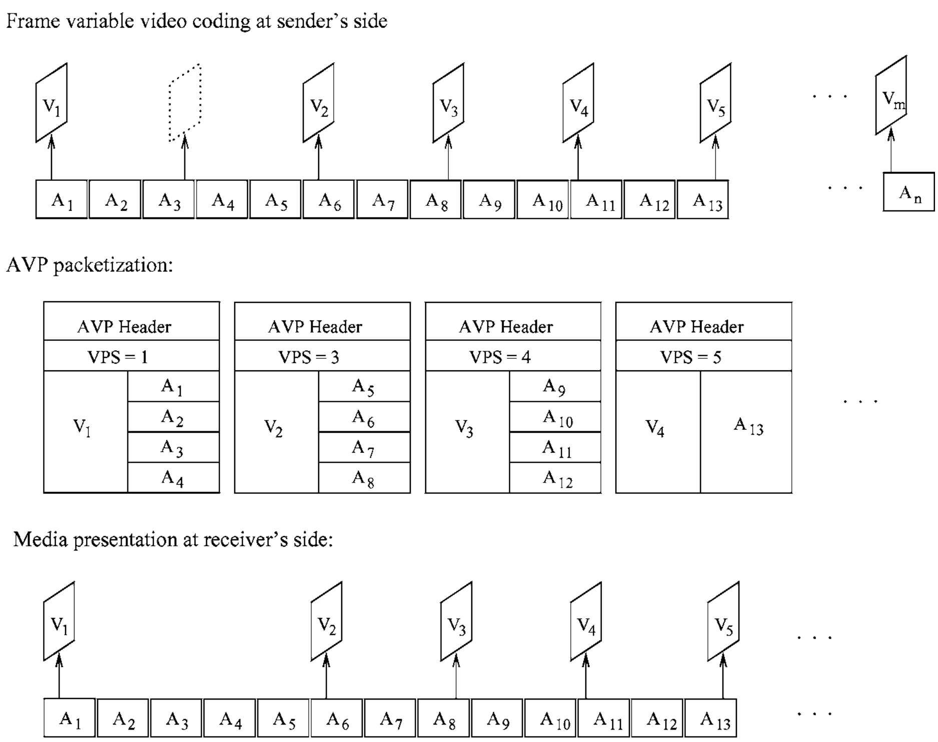

Figure 11. AVP packetization for frame variable video.

with the sequence number to achieve synchronization of multiple sources as shown in figure 12.

3.3.1. QoS control. The server can monitor the QoS according to the receiver's report. In the receiver's report, the delay and loss information of the packets is sent back such that

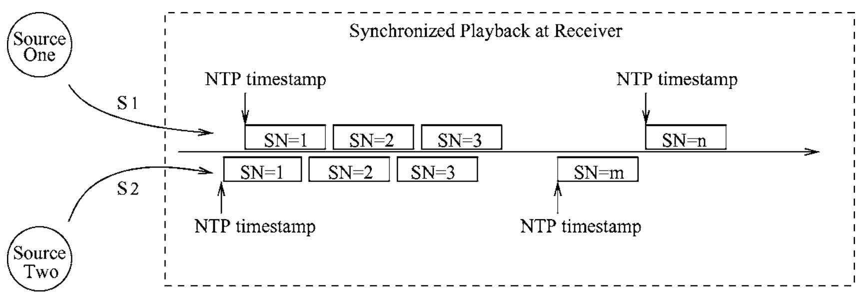

Figure 12. Synchronization for multiple sources by NTP.

the sender can adjust the packetization and transmission schemes according to the network traffic conditions. The working scheme is similar to that of RTCP [1].

3.3.2. Prioritized selective retransmission. In addition to the SR and RR fields, AVCP has an optional component of acknowledgment (NAK). NAK is sent back to indicate packet loss when the prioritized selective retransmission function of AVP is activated by the application. The content of NAK is the sequence number of the high priority packet which did not arrive on time and needs retransmission.

The basic idea of the prioritized selective retransmission scheme is to sacrifice the packets of low priority to ensure the delivery of high priority packets. When activated, the receiver will set a threshold D_T for the inter-arrival delay of high priority packets. Assuming that the previous high priority packet arrives at time T_1, if the next high priority packet has not arrived by time $T_1 + D_T$, packet loss is assumed and an NAK will be sent back. When the sender receives an NAK, it will immediately retransmit the packet and drop all the packets of the lowest priority in the following synchronized group.

Since retransmission is not desirable in most real-time applications, the users should be very careful when using the prioritized selective retransmission scheme. Nevertheless, this scheme can improve the efficiency and quality of some applications which are very sensitive to the loss of high priority packets.

4. AVP performance analysis

AVP can improve the transmission efficiency, since the timestamp is eliminated and the packet rate is reduced by combining video and audio together. In addition, the information carried by AVP header and extensive header helps to maintain good synchronization even if some packets are lost. In this section, the performance of AVP is analyzed for both synchronization and efficiency.

4.1. Synchronization performance

According to the audio-based synchronization scheme, the receiver will group the packets with similar group numbers together for playback. If there is no packet loss during the transmission, the time information can be well recovered. Packet loss can be detected by the receiver using the sequence number. If packet loss is presented, the receiver must make use of the information packed in the AVP extensive header to achieve synchronization.

According to the content of the lost packet, we classify the packet loss into three categories: video packet loss, audio packet loss, and mixed packet loss. The video packet loss here refers to the loss of the video only packet, the audio to the audio only packet, and the mixed to either the combination of the above two packets or the AVP mixed packet. The synchronization performance will be analyzed under these three situations individually.

4.1.1. Video packet loss. The effect of the video packet loss on synchronization is minor because it is audio that carries the time information. In some cases, the video packet loss will not affect synchronization at all. For example, in the 3-D wavelet compression, video packet loss will cause quality degradation, but have no effect on synchronization. Only when a synchronized group contains multiple video frames, it is difficult for the receiver to determine the positions of the remaining frames in the group. This problem can be solved by using the 4-bit AVP video extensive header. Since the play position of the video frame is explicitly shown in the extensive header, the receiver knows when to play the reconstructed video frames even with the video packet loss.

4.1.2. Audio packet loss. Since audio plays a very important role in synchronization, AVP must deal with the audio packet loss. The approach is to store redundant audio information in the AVP audio extensive header. According to the AVP packetization rules, if there are N audio packets in one synchronized group, the audio length in packet P_i, $i = 1, 2, \ldots N-1$, should be the same. Only the audio length in the last packet P_N could be different. According to the definition of field *AFN* in the audio extensive header, the audio length in packet P_N is recorded in the previous packets P_i, $i = 1, 2, \ldots N-1$ while the audio length in packets P_i, $i = 1, 2, \ldots N-1$ is recorded in P_N. Hence, if one or more packets between P_1 and P_{N-1} are lost, the receiver can still recover the time information from the extensive header of the P_N or other received packets between P_1 and P_{N-1}; if P_N is lost, the time information can be recovered from the extensive header of any packet between P_1 and P_{N-1}, as shown in figure 13. If all audio packets are lost in one synchronization group, there will be no

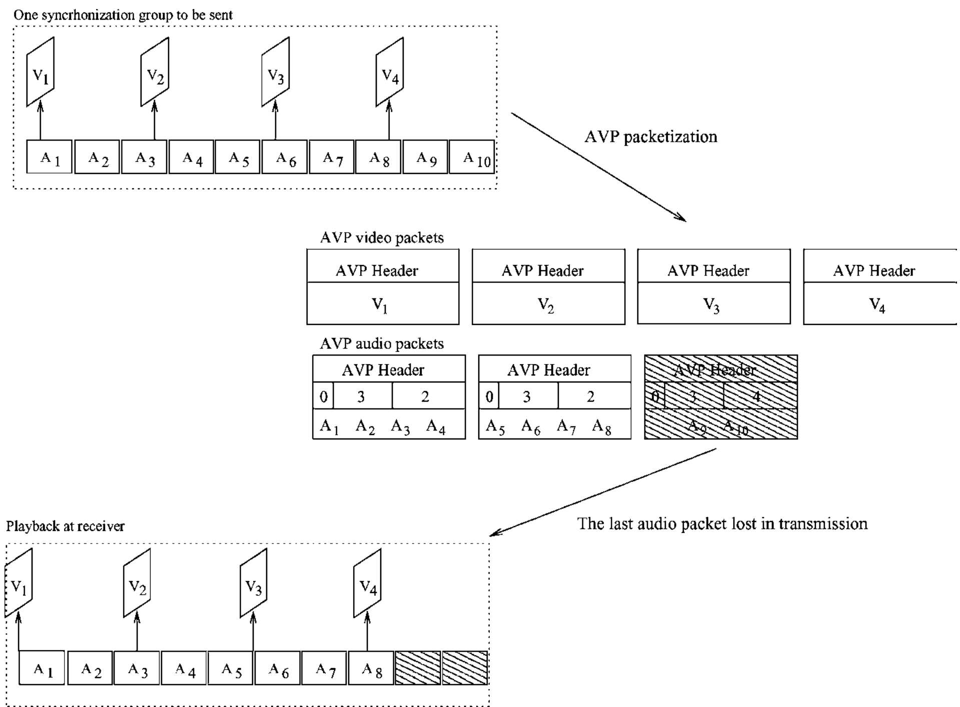

Figure 13. Synchronization performance with audio packet loss.

lip synchronization problem. The receiver can just play the video frames according to the frame rate with a silence period. As a result, synchronization is attained in any case.

4.1.3. Mixed packet loss. This situation is the combination of the above two (video and audio loss); therefore, AVP can handle it with the combination of the two approaches. For the reconstructable video frames, the receiver can always determine their relative play time with the aid of the AVP video extensive header. Also the time information can always be retrieved if any of the audio packets is received as stated above. If all the audio frames are lost but the last video frame is available, the receiver can still use the video frame rate as valid time information. If the last video frame is lost as well as all the audio packets, the total display time of this group can not be retrieved accurately. However, the lip synchronization is still achieved and only the play time of the next group is affected, which may cause some problems in interactive applications, but is quite acceptable in video streaming.

In reality, since a gap is unavoidable because of the data loss, the receiver can schedule the play time of the next group according to its buffer condition. For example, if the buffer is overflow, the next group can be played earlier. We can use an example to show how AVP handles the mixed packet loss. Assume that the video frame rate is 20 fps, and the audio frame length is 20 ms. The sender packetizes the media in the way as shown in figure 11. When the second and the fourth packet are lost during transmission, the receiver can get the audio length in the second packet from either the first or the third one, since the audio signals are of same length. Similarly, the audio length of the fourth packet can be retrieved from the extensive header in the first or third packet. Therefore, the total time information can be recovered, which is $(4 \times 3 + 1) \times 20 = 160$ ms. Figure 14 shows the result of the media presentation when the mixed packet loss occurs. The receiver may repeat the previous video frame at the play position of the lost one to get a better result.

4.2. Transmission efficiency improvement

With the audio-based synchronization and the prioritized packetization scheme, AVP increases the transmission efficiency without performance degradation. Table 2 compares the header sizes of AVP, general RTP (separated audio-video), and BMPEG, respectively. Compared with the BMPEG header, the AVP header is at least 76 bits smaller for single-source applications. Thus AVP can save 2.28 kbps if the packet rate is 30 per second. In comparison with the general RTP, the AVP header can save 64 bits in the single-source applications and only 12 bits in the multiple-source applications. It appears that the saving

Table 2. Header sizes of AVP and RTP.

Fixed AVP header	32–64 bits
Fixed RTP header	96 bits
AVP audio extensive header	8–16 bits
AVP video extensive header	4 bits
RTP BMPEG header	32 bits

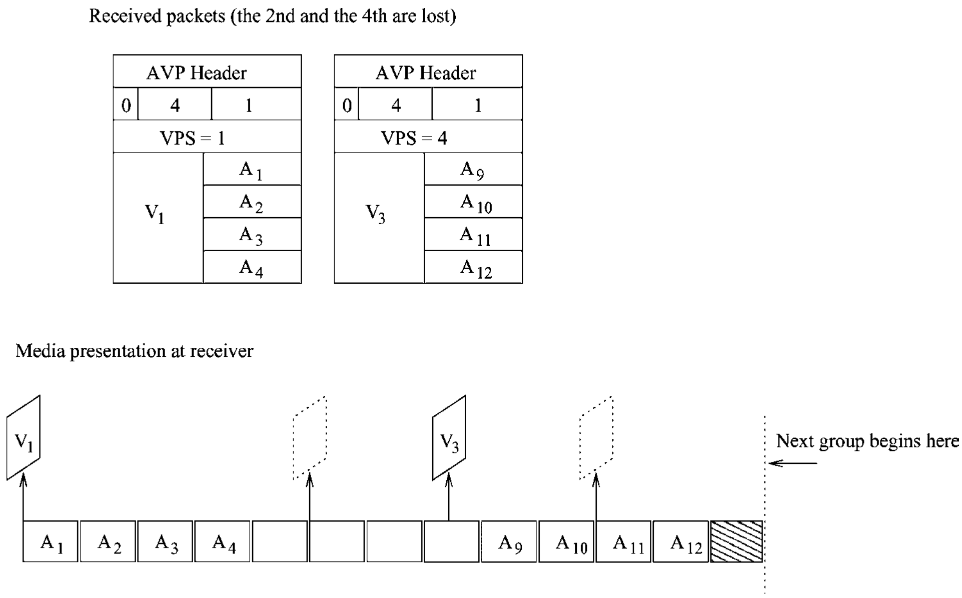

Figure 14. Synchronization performance with mixed packet loss.

is limited in the latter case. However, since AVP reduces the packet rate by packing video and audio together, it still saves a lot of bandwidth. For example, the sampling rate for the MPEG audio is 44.1 kHz, and one audio frame covers $384/44100 = 8.71$ ms. There will be about 115 audio frames in one second. Suppose RTP packetizes four audio frames into one packet, the bandwidth needed to transmit the overhead (the 36-byte RTP/UDP/IP header) of the RTP audio packets is $115 \times 36 \times 8/4 = 8.28$ kbps. Since the audio and video are packed together in AVP, the 8.28 kbps bandwidth can thus be saved. If the video packet rate is 30 per second, AVP can save $30 \times 12 + 8280 = 8.64$ kbps even for multiple source applications.

In addition to the bandwidth saving, the reduced packet rate and the single-session delivery in AVP also lead to less waiting time at the routers. As a result, the end-to-end delay of media unit is reduced which is highly desirable in real-time applications. Generally, the end-to-end unit delay can be represented by the following equation:

$$D_{\text{Unit}} = D_p + D_m + \sum_{i=1}^{m-1} \tau_{i,i+1}. \tag{1}$$

where D_p is the constant propagation delay, D_1 is the variable portion of one-way trip time for packet m in the media unit including the transmission delay and queuing delay, and $\tau_{i,i+1}$ is the interval of sending time between packet i and $i + 1$ at the source. For convenience, other parameters that will be used in the following analysis are listed in Table 3.

Table 3. Common transmission parameters.

B_V	Bandwidth of video session in RTP
B_A	Bandwidth of audio session in RTP
B_{AV}	Bandwidth of combined audio-video session in AVP
S_{V_i}	Size of the ith video packet
S_{A_i}	Size of the ith audio packet
S_{AV_i}	Size of the ith combined audio-video packet
R_V	Video source sending rate
R_A	Audio source sending rate
R_{AV}	Combined media source sending rate

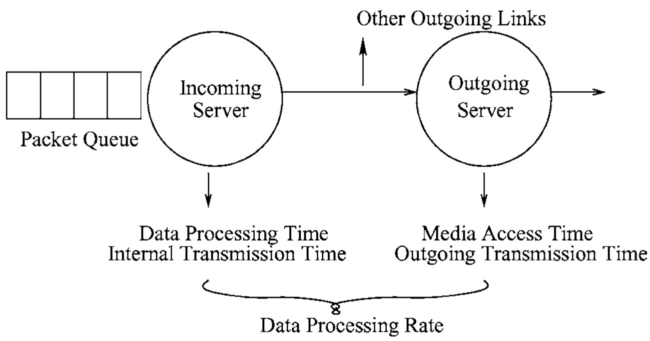

Figure 15. A two-server hop model.

For simplicity, we will use a single-hop transmission system to compare the end-to-end unit delay in RTP and AVP. Usually, a single hop can be modeled by a two-server system [25] as shown in figure 15. The time needed to pass the hop consists of two parts. One is independent of the packet size such as the data processing time at the incoming server and the media access time at the outgoing server, and the other varies with the packet size such as the internal transmission time and the outgoing transmission time. For simplicity, we define the data processing rate as the size of data (in bits) which pass the hop (including the incoming server and the outgoing server) in a time unit. Hence, this parameter takes all the above factors into consideration, and we can use it to receive a good approximation in an average sense.

Assuming the average data processing rate for hop k to be R_{h_k}, the waiting time for a packet P_m^k (the mth packet in the queue at hop k) should be:

$$\sum_{i=1}^{m-1} S_{P_i}^k \Big/ R_{h_k}.$$

where $S_{P_i}^k$ is the size for packet i at hop k and $\sum_{i=1}^{m-1} S_{P_i}^k$ is the total data size before packet P_m^k in the queue at hop k.

In RTP, two queues are formed to wait for the service of hop k since video and audio packets are sent separately. Suppose that the data size is $SS^k_{V_m}$ before the latest video packet $P^k_{V_m}$ and the data size is $SS^k_{A_n}$ before the latest audio packet $P^k_{A_n}$, the waiting time for $P^k_{V_m}$ is $SS^k_{V_m}/R_{Vh_k}$ and the waiting time for $P^k_{A_n}$ is $SS^k_{A_n}/R_{Ah_k}$, where R_{Vh_k} and R_{Ah_k} are the data process rate of the video and audio sessions at hop k, respectively. So the waiting time for the latest packet in the whole message including video and audio is:

$$\max\left(SS^k_{A_n}\big/R_{Ah_k}, SS^k_{V_m}\big/R_{Vh_k}\right).$$

In AVP, only one queue is developed. The waiting time of the latest combined video-audio packet $P^k_{AV_l}$ is $SS^k_{AV_l}/R_{h_k}$, where $SS^k_{AV_l}$ is the data size before $P^k_{AV_l}$ in the queue. Since other conditions are the same, we have:

$$\begin{cases} SS^k_{AV_l} \doteq SS^k_{V_m} + SS^k_{A_n} \\ R_{h_k} = R_{Vh_k} + R_{Ah_k}. \end{cases}$$

It is easy to prove that $\max(SS_{V^k_m}/R_{Vh_k}, SS^k_{A_n}/R_{Ah_k}) \geq SS^k_{AV_l}/R_{h_k}$. That is, the queuing delay is smaller in AVP than RTP. Since the queuing delay is the dominant factor in the end-to-end delay, the overall performance of AVP is better than that of RTP.

5. Performance evaluation

In order to evaluate the performance of our proposed protocol, a number of experiments were conducted. First of all, the delay and jitter performance of AVP and RTP are evaluated since they are the key factors affecting the perceptual quality in real-time applications. Secondly, the quality of video delivered via AVP and RTP are compared to show the impact of bandwidth saving of AVP. The following subsections will describe the two kinds of experiments and present the results.

5.1. Delay and jitter behavior

In this part, the end-to-end delays of the media unit and the synchronized media unit, respectively, are measured and compared under different traffic situations. Jitters, which reflect the difference between the interarrival delay of the media unit, will also be calculated and compared for the two protocols. Before proceeding any further, we first define the media unit and the synchronized media unit here.

- *Media unit*: One or multiple media frames which are decodable and presentable on their own. For example, in an MPEG format of I-B-B-P, the I frame is a media unit, but the P frame is not because it can not be displayed until the B frames are displayed.
- *Synchronized media unit*: A set of media units which must be presented together.

Generally, a media unit can be divided into multiple transport packets; therefore, the end-to-end delay of a media unit is $D_m = T_{\text{receive}^{\text{last}}} - T_{\text{send}^{\text{first}}}$, where D_m is the end-to-end media unit delay, $T_{\text{receive}^{\text{last}}}$ is the arrival time of the last packet of the media unit, and

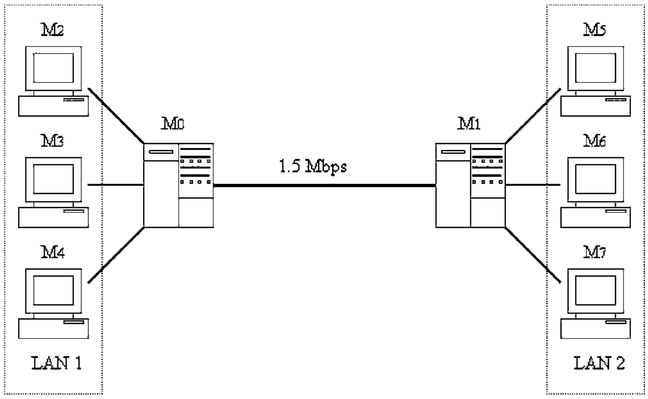

Figure 16. Network topology of simulations.

$T_{send^{first}}$ is the sent-out time for the first packet. For a synchronized media unit, the end-to-end delay includes not only the packet travel time but also the buffering time needed for synchronization. In the simulations, the buffering time consists of only the waiting time of the early arrived media unit for its associated media units.

The simulation software used in our experiments is NS2 (Version 2) [18], which is a widely used discrete event simulator in the network research. The topology is shown in figure 16. The video and audio are stored in M_3 and will be delivered to M_5. The bandwidth of the long distance link between M_0 and M_1 is 1.5 Mbps, and the propagation delay is 100 ms. Since most of the Internet traffic is caused by web browsing, we use randomly generated HTTP sessions [6] as the background traffic in our simulations.

In the simulations, we focus on the narrowband transmission because it is a realistic situation for home network users. In order to mimic the narrowband connection, we set the bandwidth of the short links between the end users and the routers as shown in figure 16 to 64 kbps. Low bit rate video and audio traffics (with parameters in Tables 4–1 and 4–2) are generated for this simulation.

The performances of AVP and RTP under both light and heavy traffic conditions (with the parameters shown in Table 5) are investigated. From figures 17 and 18, we can see that the end-to-end delay reduction of AVP is very significant, especially when the background

Table 4-1. Parameters of the low bit rate video.

Payload	Frame size (bytes/frame)	Frame rate (fps)	Ave. bit rate (kbps)
MPEG video	25359	5	22.5

Table 4-2. Parameters of the low bit rate audio.

Payload	Sampling rate (kHz)	Frame rate (ms)	Bit rate (kbps)
G.729 audio	8	10	8

Table 5. Parameters of background traffic in narrowband simulations.

	Session no.	Page no.	Pagesize	Object no.	Object size
Light	2	10	1 kB	10	1.2 kB
Heavy	100	250	1 kB	12	1.2 kB

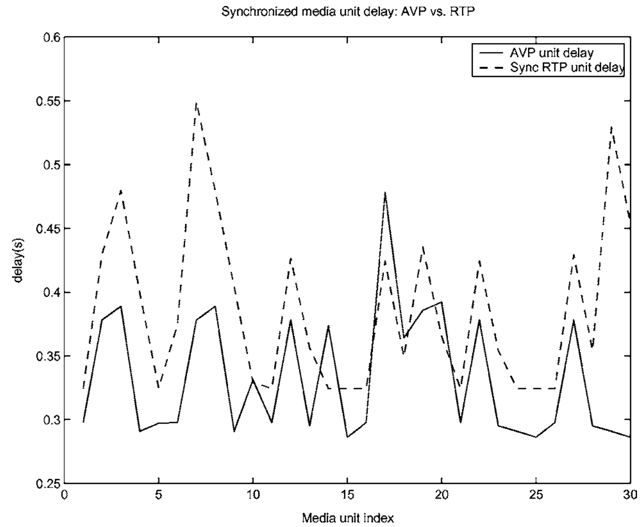

Figure 17. Synchronized media unit delay under light background traffic (64 kbps).

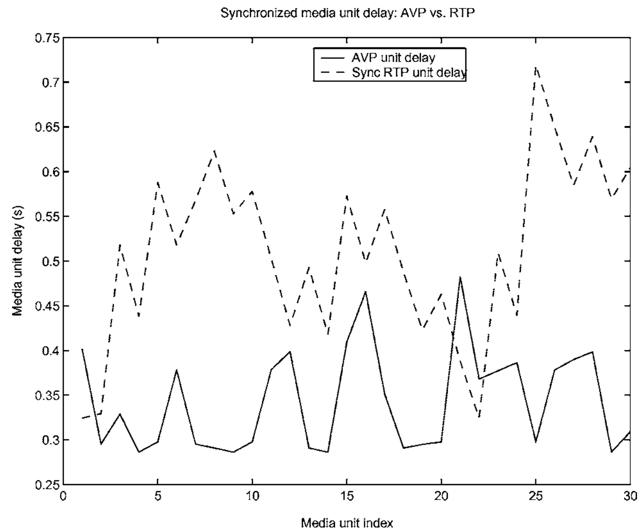

Figure 18. Synchronized media unit delay under heavy background traffic (64 kbps).

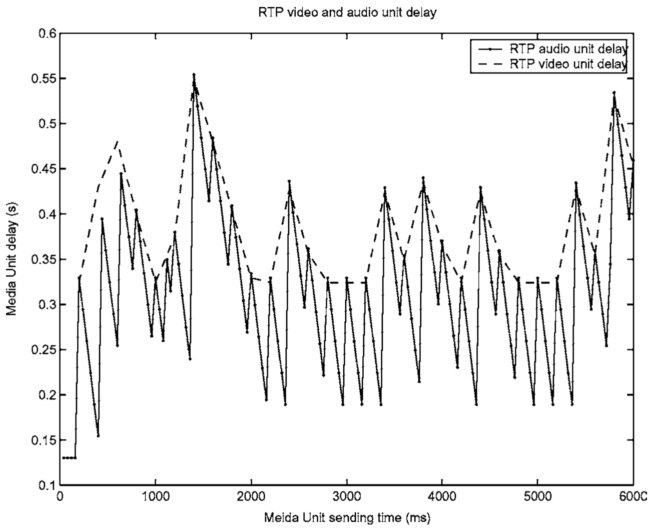

Figure 19. RTP media unit delay under light background traffic (64 kbps).

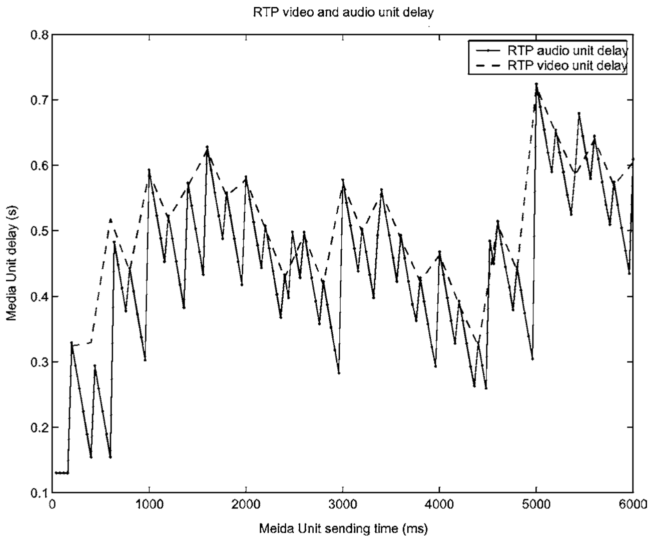

Figure 20. RTP media unit delay under heavy background traffic (64 kbps).

traffic is heavy. Figures 19 and 20 show the delays of RTP video and audio units under both light and heavy background traffic, respectively. Although the size of audio packet is small, the delay is quite long due to the waiting time at intermediate nodes. As analyzed in Section 4, the reduction of the end-to-end delay using AVP primarily comes from the less queuing delay of the packets due to the reduced packet rate by combining audio and video together. The overhead reduction also saves a lot of delivery time in narrowband applications.

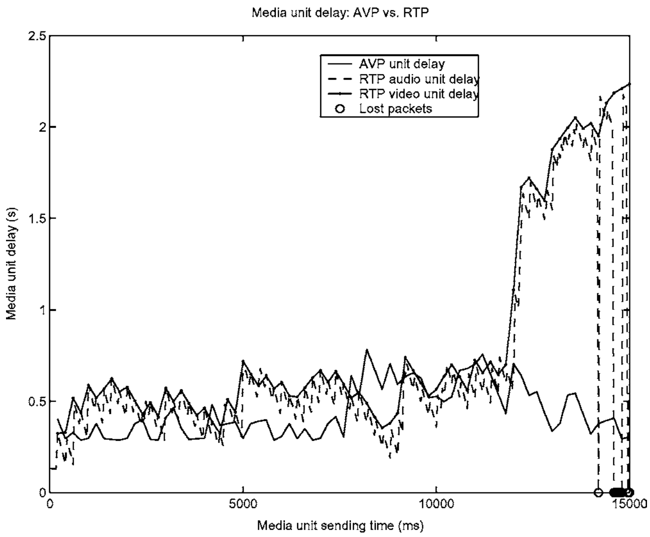

Figure 21. Media unit delay under heavy background traffic for whole simulation (64 kbps).

Figure 20 shows the beginning part of the simulation, and we can see that there is no packet loss. However, as the simulation continues, the delay of the RTP audio unit aggregates and packet loss takes place when the elapse time is around 15 s, as shown in figure 21. This phenomenon demonstrates that too many small packets will causes longer delay especially when the bandwidth is narrow.

From the above simulation results, we can see that the end-to-end delay of the AVP unit is generally shorter than that of the RTP unit, and the delay reduction is very significant in the narrowband applications. Table 6 shows the mean value of the AVP unit delay and the synchronized RTP unit delay, respectively, for a period of 15 seconds in different situations as well as the calculated jitters. From the table, we can see that AVP can save nearly 50% end-to-end delivery time, and smaller jitters are highly desirable in continuous media display. In general, AVP can provide a more efficient delivery of video-audio applications on the Internet than RTP.

5.2. Impact of bandwidth saving on video quality

In order to investigate how much improvement of video quality can be achieved by using AVP, we designed the following experiment. A compressed video bitstream is delivered

Table 6. Average media delay and jitter under different traffic conditions.

Traffic conditions	AVP unit delay (ms)	Sync. RTP unit delay (ms)	AVP unit jitter (ms)	Sync. RTP unit jitter (ms)
Narrowband w. light traffic	318.9	367.6	43.6	51.4
Narrowband w. heavy traffic	441.2	835.3	73.3	87.9

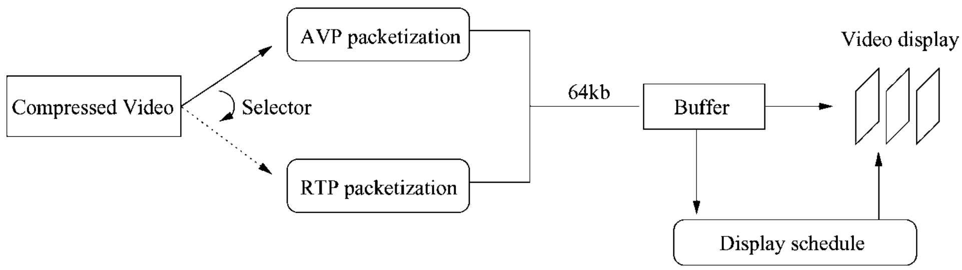

Figure 22. System diagram of video streaming simulator.

across a simulated 64 kbps channel in which either AVP or RTP can be selected as the transport protocol. At the receiver side, the protocol header will be removed and the video will be reconstructed for display (figure 22).

In the simulation, the compressed video bitstream is generated by the 3-D wavelet compression software produced in our research group [5]. Since a group of video frames (GOF) is used in the 3-D wavelet transform for compression, the packets belonging to the same GOF must be received before a deadline T_D for timely reconstruction and display, and late packets will be dropped by the receiver. Apparently, more efficient delivery causes less packet dropping, and eventually leads to better visual effect of the video at the receiver side.

Figure 23 compares the PSNR of the received video delivered by AVP and RTP, respectively. Since colored video sequences are used in our experiment, the PSNR shown in the

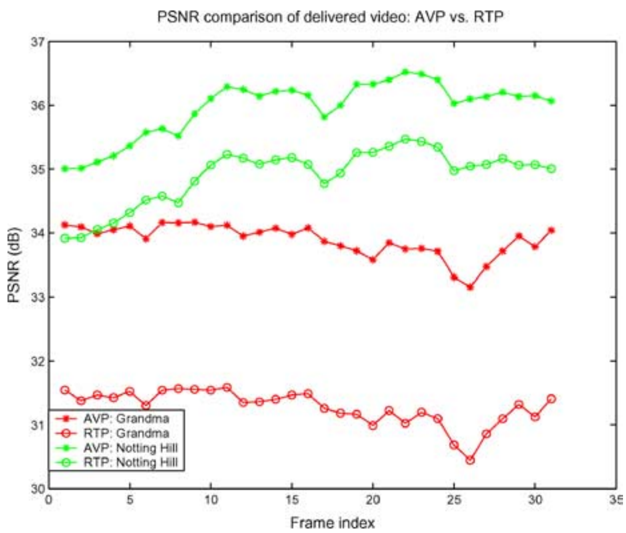

Figure 23. Frame-by-frame PSNR comparison of the video sequences delivered by AVP and RTP.

figure is calculated using the PSNRs of the YUV components as the following:

$$\mathrm{PSNR} = \frac{\mathrm{PSNR}_Y}{2} + \frac{\mathrm{PSNR}_U}{4} + \frac{\mathrm{PSNR}_V}{4}.$$

From the experiment results, the PSNR of the video delivered by AVP is one or two dB higher than that delivered by RTP.

In broadband applications where the video quality is very high, one or two dB difference can hardly be distinguished by human eyes. In narrowband applications, that difference can be very significant. Figure 24 is the original 31st video frame of the video sequence Grandma, and figures 25 and 26 are the reconstructed video frames delivered via RTP and AVP, respectively. Since the RTP header occupies a large share of bandwidth, some color information is lost and the video quality is degraded significantly (figure 25). In comparison, AVP achieves much better quality as shown in figure 26.

Figure 24. The original 31st frame in video sequence Grandma.

Figure 25. The reconstructed 31st frame delivered by RTP.

Figure 26. The reconstructed 31st frame delivered by AVP.

6. Conclusions

In this paper, we have presented a new transport protocol for multimedia communications called AVP. AVP adopts a novel synchronization scheme which uses audio signal as the time reference. While achieving easier synchronization, this scheme eliminates the timestamp and significantly reduces the header size. Also fewer control packets are needed for delivering the same amount of data. As a result, the transmission efficiency is improved.

It should be noted that eliminating the timestamp in the transmission process will not affect the functions which the timestamp has to play on the receiver side. This is because the latter can use a calculated timestamp using the audio reference plus the NTP timestamp which is still preserved. As a result, the receiver side can still schedule the display time of the audio-video packets and compute the interarrival jitter for QoS control. In addition, the AVP video extensive header can work with the group number to maintain the framing function of the RTP timestamp when packet loss occurs.

Another attractive advantage of AVP is the prioritized packetization scheme. With the prioritized packetization scheme, different priorities can be assigned to packets at both the transport and the network layers (assume IPv6 is used). Based on the prioritized packetization scheme, a prioritized selective retransmission scheme can be deployed to improve the transmission reliability of the high priority packets in some applications.

To implement the audio-based synchronization scheme, we designed the AVP fixed header and the extensive header, respectively. The AVP fixed header provides necessary information for synchronization and packet loss detection, etc., and the AVP extensive header is designed to achieve better error recovery. With the extensive header, AVP can perform quite well when packet loss occurs.

To evaluate the performance of AVP, a series of simulations are conducted. The end-to-end unit delay of AVP is measured and compared with that of RTP under different traffic conditions in narrowband applications. Simulation results show that AVP can achieve lower end-to-end unit delays for media delivery under various traffic conditions. Furthermore, the visual quality of the received video delivered by AVP is better because the overhead is smaller and more bits are used for delivering media data. In conclusion, AVP can achieve a highly efficient multimedia delivery with a better overall quality than the RTP protocol, which is very attractive in low bit rate streaming applications.

AVP is not to replace RTP completely (RTP is advantageous in interactive applications such as videoconference), but to provide a better solution for streaming stored video on the Internet. Using AVP in many emerging applications such as video on demand is more efficient than using RTP. However, there are two major barriers for wide adoption of AVP. The first is the implementation cost. Although the cost of the hardware implementation of AVP will not be much higher since the complexity of AVP is about same as RVP, people are reluctant to replace a functional protocol with any new one even if the latter has better performance due to the overall cost of replacement. Secondly, streaming multimedia on the Internet has not become a widespread application of the Internet. Hence there is no urgent need to implement a new protocol designated for streaming. It is our belief that with continuing increase of various streaming applications on the Internet, AVP can be adopted to achieve better streaming quality of multimedia.

Acknowledgment

This work was supported by the Air Force Research Laboratory and the Dayton Area Graduate Studies Institute.

References

1. I. Busse, B. Deffner, and H. Schulzrinne, “Dynamical QoS control of multimedia applications based on RTP,” Computer Communications, Vol. 19, pp. 49–58, 1996.
2. S. Casner and V. Jacobson, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, RFC 2508, Internet Engineering Task Force, Feb. 1999.
3. G. Cash, M. Civanlar, and B. Haskell, RTP Payload Format for Bundled MPEG, RFC 2343, Internet Engineering Task Force, 1998.
4. D. Clark and D. Tennenhouse, “Architectural considerations for a new generation of network protocols,” in Proc. ACM Sigcomm 90, 1990.
5. J. Dong and Y.F. Zheng, “An innovative framework for 3-D wavelet video streaming,” in Proc. SPIE AeroSense’ 2002, April 2002, to appear.
6. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, Hypertext Transfer Protocol—HTTP/1.1. RFC 2068, Internet Engineering Task Force, Jan. 1997.
7. H.225.0, Media Stream Packetization and Synchronization for Visual Telephone Systems on Non-Guaranteed Quality of Service LANs, ITU-T Rec., 1996.
8. H.263+, Video Coding for Low Bitrate Communications, ITU-T Rec., Jan. 1998.
9. H.323, Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service, ITU-T Rec., 1996.
10. M. Handley and C. Perkins, Guidelines for Writers of RTP Payload Format Specifications. RFC 2736, Internet Engineering Task Force, Dec. 1999.
11. C. Huitema, IPv6: The Next Generation Internet Protocol, Prentice Hall, 1998.
12. ISO/IEC 13818-2, Information technology—Generic Coding of Moving Pictures and Associated Audio Information: Video, March 1995.

13. A. Kantarci and T. Tunali, “The design and implementation of a streaming application for MPEG videos,” in Proc. 2000 IEEE International Conference on Multimedia and Expo, Vol. 2, 2000, pp. 1021–1024.
14. B.J. Kim, Z. Xiong, and W.A. Pearlman, “Low bit-rate scalable coding with 3-D set partitioning in hierarchical tree (3-D SPIHT),” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, pp. 1374–1387, 2000.
15. F. Kossentini, TMN 8 (H.263+) Encoder/Decoder, Signal Processing and Multimedia Group, University of British Columbia, http://spmg.ece.ubc.ca/h263plus/, 1998.
16. I.K. Levy and R. Wilson, “Three-dimensional wavelet transform video coding using symmetric codebook vector quantization,” IEEE Transactions on Image Processing, Vol. 10, pp. 470–475, 2001.
17. S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1998.
18. S. McCanne and S. Floyd, The UCB/LBNL Network Simulator, University of California, Berkeley, http://www-mash.cs.berkeley.edu/ns/.
19. D. Mills, Network time protocol (v3). RFC 1305, Internet Engineering Task Force, April 1992.
20. S. Palacharla, A. Karmouch, and S. Mahmoud, “Design and implementation of a real-time multimedia presentation system using RTP,” in Proc. The 21th Annual International Computer Software and Applications Conference, 1997, pp. 376–381.
21. B. Paul, D. Gibbon, G. Cash, and M. Civanlar, “Vtondemand: A framework for indexing, searching and on-demand playback of RTP-based multimedia conferences,” in Proc. 1999 IEEE 3rd Workshop on Multimedia Signal Processing, 1999, pp. 59–64.
22. R. Steinmetz and C. Engler, “Human perception of media synchronization,” Tech. Rep. 43.9310, IBM European Networking Center, Aug. 1993.
23. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications. RFC 1889, Internet Engineering Task Force, 1995.
24. J. Shapiro, “Embedded image coding using zero-trees of wavelet coefficients,” IEEE Transactions on Image processing, Vol. 41, pp. 3445–3462, 1993.
25. T. Talley and K. Jeffay, “A general framework for continuous media transmission control,” in Proc. 21st IEEE Conference on Local Computer Networks Minneapolis, MN, Oct. 1996, pp. 374–383.
26. J.Y. Tham, S. Ranganath, and A.A. Kassim, “Highly scalable wavelet-based video codec for very low bit-rate environment,” IEEE Journal on Selected Areas in Communications, Vol. 16, pp. 12–27, 1998.

Jianyu Dong received the B.E. and M.S. degree from the University of Science and Technology of China (Hefei, China) in 1995 and 1998, respectively, and the Ph.D. degree from The Ohio State University, Columbus, in 2002, all in electrical engineering.

In 2002, she joined the faculty of California State University, Los Angeles, where she is currently an Assistant Professor in the Department of Electrical and Computer Engineering. While continuing research on video compression and streaming, she is also involved in the NASA-sponsored project in the SPACE laboratory, and is leading the research activities of astronomical image processing, compression and transmission.

Chao He received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1995, and the M.S. degree in electrical engineering in 2000 from The Ohio State University (OSU), Columbus, where is currently working toward the Ph.D. degree. His research interest includes audio/video compression and streaming.

Yuan F. Zheng received the MS and Ph.D. degrees in *Electrical Engineering* from *The Ohio State University*, Columbus, Ohio in 1980 and 1984, respectively. His undergraduate education was received at *Tsinghua University*, Beijing, China in 1970. From 1984 to 1989, he was with the Department of Electrical and Computer Engineering at Clemson University, Clemson, South Carolina. Since August 1989, he has been with The Ohio State University, where he is currently Professor and Chairman of Electrical and Computer Engineering. Professor Zheng's research interests include two aspects. One is in *wavelet transform* for image and video compression for internet and satellite communications. Current efforts focus on content-based compression, 3D wavelet transformation, video object tracking, and content-based retransmission in Internet communications. The other is in robotics which includes robots for biological applications, multiple robots coordination, legged robots, human-robot coordination, and personal robotics. He is currently on the Editorial Board of *International Journal of Multimedia Tools and Applications*, on the Editorial Board of *Autonomous Robots*, an associated editor of the *International Journal of Intelligent Automation and Soft Computing*, on the Editorial Board of *International Journal of Intelligent Control and Systems*, and on the Editorial Board of *International Journal of Control, Automation, and Systems.*

Professor Zheng was Vice-President for Technical Affairs of the IEEE Robotics and Automation Society from 1996 to 1999. He was an associate editor of the IEEE Transactions on Robotics and Automation between 1995 and 1997. He was the Program Chair of the 1999 IEEE International Conference on Robotics and Automation, held in Detroit, MI, on May 10–15, 1999. Professor Zheng received the Presidential Young Investigator Award in 1986.

Robert L. Ewing began his career in the Propulsion Laboratory at Wright Patterson AFB during the early 1970's with the development of jet engine control systems and the initial control system used on the F-15 aircraft. In the middle part of the 70's, he worked with the University of Cincinnati's Medical School in the area of electronic control & regeneration of peripheral (sciatic) nerves used in walking, and continued this study at the University of Edinburgh, Scotland. From 1977 to 1982, he held the position of a medical research scientist at the Aerospace Medical Research Laboratory, in the Biodynamic Effects Division (where the word, "Bionics", historically originated from a conference held there). He worked to develop the pilot's analog & digital flight control systems and aircraft ejection systems for low-level, high-speed flight. Also, during this time, he consulted with Wright State University's Medical Research & Engineering Department, in order to start its newly developed bionics & bioengineering area. In 1982, he became an instructor, for the Army, at the Air Force Institute of Technology (AFIT) & an adjunct instructor at Wright State University. During his work at the *AFIT*, he developed many of the early short courses and classes in Robotics, Digital Control, Artificial Intelligence, Neural Nets, Database Systems, Low Observables (Radar), Navigation & Guidance Systems, Microprocessor Design and Microelectromechanical Devices (MEMS). In 1993, he started working at Wright Laboratory's Solid State Electronic Devices Directorate in the area of hardware description language (*VHDL*) for VLSI synthesis. Currently, he is directing the Computer Engineering Research Consortium (*CERC*) of local universities in the area of mixed-signal design. He is working towards the development and use of hardware description language for mixed-signal design and synthesis (*VHDL-AMS*). He holds engineering degrees, with BSEE (University of Cincinnati) & Ph.D. in Electrical Engineering (University of Dayton), and a physics degree, with MS (University of Cincinnati). He has been a registered Professional Engineer (PE) with the State of Ohio since 1984, and is currently an adjunct professor at *AFIT*.