****

TITLE: Encapsulation of Real-Time Data Including RTP Streams over ATM

****

SOURCE: Alexander Fraser, Peter Onufryk and K. K. Ramakrishnan
AT&T Labs. Research
180 Park Avenue
Florham Park, N. J. 07932
Tel: (973) 360-8212,8766
Fax: (973) 360-8871
{pzo, kkrama}@research.att.com

****

DISTRIBUTION: ATM Forum Technical Committee
SAA Working Group

****

DATE: February 1998 (Anaheim)

****

ABSTRACT: This contribution proposes a format for carrying voice and other real-time streams over ATM. The encapsulation format is extensible and has the potential to support a wide range of real-time applications. One of our primary concerns is the need to support real-time data that is encapsulated with the RTP/UDP/IP protocol stack. We strictly use the existing AAL5 adaptation layer. This contribution is being submitted to the SAA working group.

****

****

Introduction

There is considerable work ongoing on the transport of real-time streams over both ATM networks and IP-based Internetworks. In particular, there has been work in the ATM Forum for carrying packetized voice, using a variety of ATM Adaptation Layers. AAL5 has been used for encapsulating voice in the Voice over ATM to the Desktop activity [4]. Also proposed is another new adaptation layer, AAL2, for carrying packet voice. The latter activity is focused on multiplexing several voice stream over an ATM Virtual Circuit. Carrying real-time streams over IP has also been considered. The real-time transport protocol, RTP [3], has been used for encapsulating the real-time data samples. Recently, ATM Forum contribution 97-0948 [5] addressed the issue of efficiency of voice stream packetization using RTP and carrying it over ATM networks. The group considered avoiding the RTP layer (as per the meeting minutes). One of the concerns was the consequence of the removal of the RTP layer.

Our focus in this contribution is to simultaneously seek efficiency, and more importantly, provide flexibility while carrying real-time streams over ATM. Our objective is to carry an individual stream over an ATM VC using a simple, yet flexible encapsulation format. Our encapsulation is based on the following objectives.

1. A uniform way to carry different types of real-time streams (i.e., voice, video or others) over ATM.
2. An encapsulation that uses existing adaptation layers ***without modification***. This enables use of existing hardware.
3. Recognizing that the streams need to communicate other data than just the real-time sample data on the same VC. For example, frames with additional headers and a means to send control information need to be supported. Our encapsulation provides a simple means to enable this.
4. Several compression techniques for real-time streams desire to manage losses by using an indication of loss, but do not require retransmissions. An encapsulation technique that enables identification of losses is desirable. We provide for a sequence number to identify the real-time sample to aid in such loss identification.

This contribution focuses on carrying compressed, packetized voice. The overhead of headers, especially with RTP/UDP/IP over a wide-range of link speeds is a concern. The small payload needed to minimize packetization delay requires as efficient an encapsulation as possible. Audio, especially telephony, desires to keep delay as small as possible. There are many things that can and need to be done to minimize delay. However, in this context, we are primarily concerned about packetization delay. For a variety of reasons, including the need to support different voice compression schemes, we assume that the payload size may be 40 bytes or less for telephony. However, our encapsulation techniques are not constrained by this payload size. We just optimize for this payload size.

Our contribution discusses how to carry real-time samples over AAL5, using a “real-time” AAL5 encapsulation. The function of encapsulation may sit just over the adaptation layer in the model of ATM layers. Our encapsulation allows us to carry real-time samples efficiently. We focus especially on carrying voice in this contribution. In addition, our encapsulation allows ATM to carry RTP/UDP/IP packets equally efficiently, by exploiting the ideas of header compression. We model the ATM Virtual Circuit as a point-to-point link for the purposes of header compression. By using a “header extension” bit in the AAL5 encapsulation, we achieve the desirable flexibility to carry other information in the AAL5 frame when necessary over the same Virtual Circuit.

Encapsulation of Real-Time Data

We would like to transport real-time data over an ATM network as efficiently as possible. We recognize that the real-time stream may be generated from a simple ATM end-systems device such as a packet telephone or it may be generated from a computer system that has an IP protocol stack in it. For ease of exposition, we will refer to the service offered by the ATM network for simple ATM end-systems as ATM-real-time service, and that for IP-based real-time streams as IP-real-time service. For example the IP-stream may be generated from a LAN-based packet telephone. The voice generated by an ATM end-system is called ATM-voice. Correspondingly, voice generated from an IP end-system using RTP/UDP/IP headers is called IP-voice in this contribution. The point where the encapsulation is performed is called the NIU (standing for the ATM Network Interface Unit) in this contribution. The compression of the RTP/UDP/IP headers is also performed at the NIU.

When ATM-voice is received at the NIU, we assume that it has already been encoded and packetized. Typically, IP-voice is also likely to be compressed voice, and it appears somewhat increasingly likely that it would be received at the NIU as a packet encapsulated with RTP/UDP/IP headers. The compression is performed at the source itself.

We assume that voice samples that are 40 bytes in length would be the appropriate unit to be transported over the access link. Our desire is to have a common encapsulation of voice, both for ATM-voice as well as IP-voice, so that the we can minimize any additional complexity arising from carrying different types of voice over the ATM network. We also assume that the compression techniques used (G.728, G.711) can take advantage of indication of loss/mis-ordered delivery of voice packets. Hence a sequence number of reasonable length would be useful. Mis-ordering is only an issue if the packet initially flows over an UP network.

Our desire is to carry a voice packet with a payload of 40 bytes in a single ATM cell whenever possible, and as far as possible in the ATM network. There are several pieces of relatively static information, such as coding type that may be negotiated end-end when the VC is setup. For ATM-voice, subsequent to the VC being setup, only the voice data flows over the VC. We seek to use AAL5 for framing the voice payload, so that we can exploit error-checking hardware that already exists for carrying packet data end-to-end. Furthermore, motivated by the need to use a common encapsulation for both ATM-voice and IP-voice, we need to carry variable length payloads as well (for RTP packets). The AAL5 adaptation trailer carries a length field that allows us to support variable length voice payloads easily.

ATM-Voice Encapsulation Over an ATM Network

The ATM Forum has been examining different forms of encapsulation of voice to be carried over ATM networks. One of the earlier encapsulation formats was AAL1, where 47 bytes of payload may be carried in a single cell. There was a 3 bit sequence number, and a 4 bit checksum, along with a single bit to convey the residual time-stamp information. We felt that, while this was probably a suitable format for PBX-PBX communication. However to meet the need to carry compressed voice of potentially more than one compression (G.728, G.711), it is desirable to stay with a 40 byte voice payload, whenever possible. The ATM Forum also has a AAL5 encapsulation format, as part of the Voice over ATM to the Desktop standardization effort. While this allowed a 40 byte payload, it does not yet allow a sequence number field, which we believe is critical for provide an indication of loss or mis-ordering to the decoder. As a result, we adapt the format from the Voice over ATM to the Desktop work.

The format we suggest for encapsulating ATM-voice over AAL5 includes a sequence number field, in the User-User field of the AAL5 frame. We believe that it is desirable that the encapsulation that we arrive at here meet the needs not only of real-time voice, but also of other real-time media streams. Therefore, it is not really necessary to constrain the payload to be 40 bytes -- it should just be an optimization for those real-time streams (such as voice) that have a payload of less than or equal to 40 bytes to be carried as a single cell. Furthermore, even for these streams, it may be useful to have an option to communicate other types of information along with the data flow. For example, we may want to communicate inter-media synchronization information, or timestamps along with the data stream on occasion. As a result, we include a bit in the trailer to allow us to indicate that preceding bytes in the AAL5 frame are not just the simple 40 bytes of voice. Simplicity suggests to have a single bit, called the *extension bit*, and include the additional appropriate information as headers appended to the payload. The length field allows us to transmit arbitrary size AAL5 frames, when required (although for ATM voice we would be predominantly using the 40 bytes of payload).

The extension bit allows us to indicate that the payload part of the AAL5 frame has additional information beyond the simple data payload. This allows us to define end-end application specific headers to convey additional information between the end systems participating in the real-time information exchange. When the trailer bit is 0 (as it would be for carrying the simple ATM-voice payload of 40 bytes) and if we chose to not use the sequence number (we would very much like to use the sequence number field), then the encapsulation is identical to that specified by the ATM Forum's Desktop VTOA group. The trailer bit would be set to 1 only if we carry other information in the packet that requires the remote destination to interpret the AAL5 payload further for headers. We will call this a Real-Time AAL5 encapsulation format from now on for ease of exposition.

	Voice Samples	Extension Bit	Seq #	MBZ	Length	CRC
Octets:	40	1-bit	7-bits	1	2	4

Figure 1. Real-Time AAL5 Encapsulation

IP Voice Encapsulation Over an ATM Network

We would like to smoothly carry voice generated from PCs over the ATM backbone, requiring little or no distinctive processing compared to carrying ATM-voice. We assume that the voice generated from PCs is likely to be delivered by an application that is running over RTP, and that in turn runs over UDP and IP. As a result, the voice payload would be encapsulated with RTP/UDP/IP headers. Typically, the fixed RTP/UDP/IP headers are 40 bytes, with 12 bytes for the RTP header and 8 bytes for the UDP header and 20 bytes for the IP header. The UDP and IP header is shown in Figure 2 while the RTP header is shown in Figure 3.

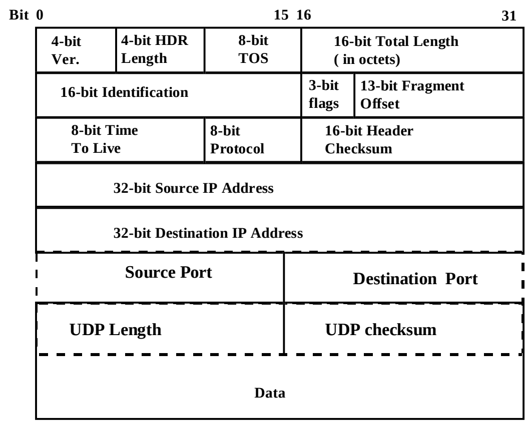

Figure 2. UDP and IP Headers

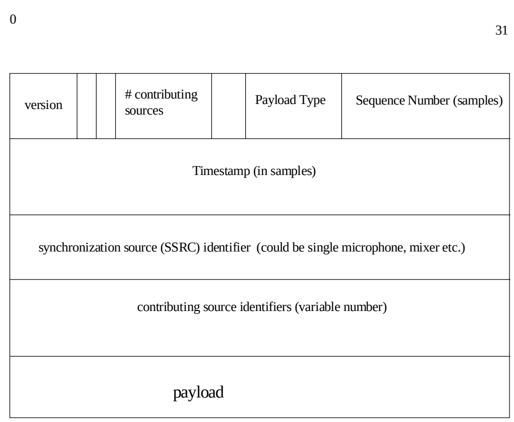

Figure 3. RTP Header

The Internet Engineering Task Force (IETF) has been working on compressing the RTP/UDP/IP headers when going over a point-to-point PPP link [1,2]. Since a virtual circuit in an ATM network is likely to look like a point-to-point link, we seek to carry the compressed packet all the way across the ATM network till we arrive at a gateway to RTP where decompression occurs. This has the added advantage of not wasting bandwidth on the ATM network, carrying uncompressed higher-layer headers over the ATM network, when it is not necessary. This recovers a substantial part of the cell-tax that is paid when carrying packets over an ATM network. When the RTP payload is of the order of 40 bytes (for example, when RTP is trying to carry 5 milliseconds of G.711 data or 20 milliseconds of G.728 data), then we would like to fit the compressed packet into 1 cell. This has the desirable characteristic of having a common encapsulation format for both ATM voice and IP voice over the ATM backbone.

It is envisioned that an RTP compressor/de-compressor function would be added to enable efficient IP voice over an ATM network. The location of this function in an ATM end system is shown in Figure 4. The compressor/de-compressor function may also be added to routers which interface IP networks to ATM networks. This is shown in Figure 5.

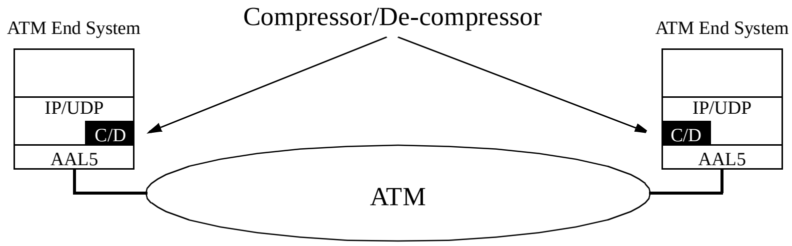

Figure 4. Compressor/De-compressor in an ATM End System

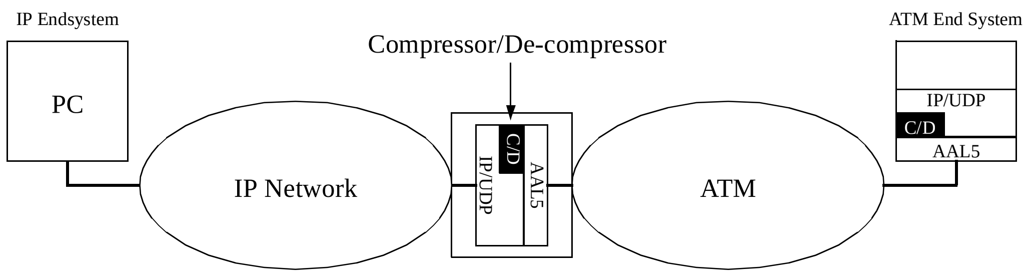

Figure 5. Compressor/De-compressor in an IP/ATM Router

In both routers and ATM end systems the compressor/de-compressor performs four functions. The first is to snoop IP traffic to determine when a new RTP flow has to be established. When a new RTP flow is detected, a new ATM VC is created to carry the compressed voice traffic between a compressor and a de-compressor. A negotiation is then performed between the compressor and de-compressor to establish initial state information required for compression and decompression of the RTP/UDP/IP headers. Once the VC and initial state are established, the compressor simply maps RTP/UDP/IP headers into the compressed format when sending RTP voice into the ATM network, and the de-compressor performs the reverse function when RTP voice is received from the ATM network. Finally, the compressor/de-compressor detect when an RTP flow is no longer active and tear down the corresponding ATM VC.

The format of the compressed RTP/UDP/IP header is shown in Figure 6. We would like to have a similar encapsulation, at the AAL5 frame level, for both uncompressed and compressed RTP/UDP/IP headers, so that the work performed at the NIU for this mapping to our Real-Time AAL5 encapsulation is minimal. The compressed RTP/UDP/IP packet with a the RTP payload would appear as shown in Figure 7 for the Real-Time AAL5 encapsulation.

0/**1**	1	Generation	16-bits for session context ID	M	S	T	I	Seq # (4-bits)
UDP checksum			RTP data					

Figure 6. Compressed RTP/UDP/IP Packet

	RTP payload	Extension Bit	Seq #	MBZ	Length	CRC
Octets:	Arbitrary (1 to 65535 octets)	1-bit	7-bits	1	2	4
		0				

Figure 7. Real-Time AAL5 Encapsulation of Compressed RTP/UDP/IP Packet

The extension bit would be set to 0 for the packet with compressed headers. The only portion of the compressed header that the NIU would extract would be the sequence number in the compressed RTP header (which is 4 bits). This would be placed in the 7 bit sequence number field in the AAL5 trailer. Since there is a 4 byte CRC in the AAL5 trailer, it would be ideal if the source did not in fact use the UDP checksum for error checking purposes. Of course, we do not preclude the use of the UDP checksum, if it is considered necessary for error protection beyond the point where the CRC is checked, but before the RTP receiver. The same Real-Time AAL5 encapsulation may be used to carry the uncompressed RTP/UDP/IP packet with the payload as well. The extension bit is set to 1 to convey the fact that this is in fact an uncompressed packet. This is shown in Figure 8.

	Full RTP/UDP/IP payload	Extension Bit	Seq #	MBZ	Length	CRC
Octets:	Arbitrary (1 to 65535 octets)	1-bit	7-bits	1	2	4
		1				

Figure 8. Real-Time AAL5 Encapsulation of Uncompressed RTP/UDP/IP Packet

Thus, the same encapsulation is used to carry both compressed as well as uncompressed RTP/UDP/IP packets as well as for carrying ATM-voice or other native ATM Real-Time streams. We believe that it is a highly desirable feature to have a common encapsulation format.

Open Issues

There are several issues that this contribution has not addressed. We recognize that these are open issues that have to be addressed in further contributions. Briefly some of the open issues are:

1. Setting up a VC for a given RTP/UDP/IP flow. When is the VC set up? When the router has the NIU function described here, it would have to “peek” into a packet to determine an existing/new flow and then setup a VC for a new flow.
2. When an end-system or router has an NIU performing the header compression and encapsulation, we also need to determine the ATM address of the remote end-node to which the setup is issued. This of course is not new functionality: this is the address resolution function needed for IP-ATM interworking.
3. Protocols need to be defined for exchanging information between the RTP header compressor and decompressor. This is typically done on either end of a PPP link. We need to have this function also over an ATM VC.
4. We also need to address the usual issues of managing and asking for, Quality of Service over an ATM VC for an RTP/UDP/IP flow.
5. We need to examine issues of interoperation with the AAL5 ATM Voice to the Desktop standardized encapsulation. There are a few possible alternatives.
    - If ATM signaling allowed for negotiation, the two ends of an ATM VC could choose the “least common denominator” for the encapsulation choice.

- A receiving end-system that is capable of, and is using, RT-AAL5 could insert a "dummy" sequence number.

6. We need to examine the interoperation between H.323 terminals for all the other functions, and the impact of our encapsulation technique. We believe that our encapsulation should in fact not change, and only make it simpler to deal with, these issues.
7. We also need to examine interoperation with AAL2.

References

1. S. Casner and V. Jacobson. Audio/Video Transport Working Group, Internet Engineering Task Force Internet Draft, draft-ietf-avt-crtp-03.txt, July 11, 1997 (expires 12/97)

2. V. Jacobson. TCP/IP Compression for Low Speed Serial Links, RFC 1144.

3. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for real-time Applications, RFC 1889.

4. Voice and Telephony over ATM to the Desktop, af-vtoa-0083.000, ATM Forum, May 1997.

5. Audet: H.323 Voice Stream Packetization on ATM, ATM Forum Contribution 97-0948, Dec. 1997. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links