RTP Payload Format for
the Adaptive TRansform Acoustic Coding (ATRAC) Family

Abstract

   This document describes an RTP payload format for efficient and
   flexible transporting of audio data encoded with the Adaptive
   TRansform Audio Coding (ATRAC) family of codecs. Recent enhancements
   to the ATRAC family of codecs support high-quality audio coding with
   multiple channels. The RTP payload format as presented in this
   document also includes support for data fragmentation, elementary
   redundancy measures, and a variation on scalable streaming.

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

   The ATRAC family of perceptual audio codecs is designed to address
   numerous needs for high-quality, low-bit-rate audio transfer. ATRAC
   technology can be found in many consumer and professional products
   and applications, including MD players, CD players, voice recorders,
   and mobile phones.

   Recent advances in ATRAC technology allow for multiple channels of
   audio to be encoded in customizable groupings. This should allow for
   future expansions in scaled streaming to provide the greatest
   flexibility in streaming any one of the ATRAC family member codecs;
   however, this payload format does not distinguish between the codecs
   on a packet level.

   This simplified payload format contains only the basic information
   needed to disassemble a packet of ATRAC audio in order to decode it.
   There is also basic support for fragmentation and redundancy.

2. Conventions Used in This Document

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [4].

3. Codec-Specific Details

   Early versions of the ATRAC codec handled only two channels of audio
   at 44.1 kHz sampling frequency, with typical bit-rates between 66
   kbps and 132 kbps. The latest version allows for a maximum of 8
   channels of audio, up to 96 kHz in sampling frequency, and a lossless
   encoding option that can be transmitted in either a scalable (also
   known as High-Speed Transfer mode) or standard (aka Standard mode)
   format. The feasible bit-rate range has also expanded, allowing from
   a low of 8 kbps up to 1400 kbps in lossy encoding modes.

   Depending on the version of ATRAC used, the sample-frame size is
   either 512, 1024, or 2048 samples. While the lossy and Standard mode
   lossless formats are encoded as sequential single audio frames,
   High-Speed Transfer mode lossless data comprises two layers -- a
   lossy base layer and an enhancement layer.

   Although streaming of multi-channel audio is supported depending on
   the ATRAC version used, all encoded audio for a given time period is
   contained within a single frame. Therefore, there is no interleaving
   nor splitting of audio data on a per-channel basis with which to be
   concerned.

4. RTP Packetization and Transport of ATRAC-Family Streams

4.1. ATRAC Frames

   For transportation of compressed audio data, ATRAC uses the concept
   of frames. ATRAC frames are the smallest data unit for which timing
   information is attributed. Frames are octet-aligned by definition.

4.2. Concatenation of Frames

   It is often possible to carry multiple frames in one RTP packet.
   This can be useful in audio, where on a LAN with a 1500-byte MTU, an
   average of 7 complete 64 kbps ATRAC frames could be carried in a
   single RTP packet, as each ATRAC frame would be approximately 200
   bytes. ATRAC frames may be of fixed or variable length. To
   facilitate parsing in the case of multiple frames in one RTP packet,
   the size of each frame is made known to the receiver by carrying
   "in-band" the frame size for each contained frame in an RTP packet.
   However, to simplify the implementation of RTP receivers, it is
   required that when multiple frames are carried in an RTP packet, each
   frame MUST be complete, i.e., the number of frames in an RTP packet
   MUST be integral.

4.3. Frame Fragmentation

   The ATRAC codec can handle very large frames. As most IP networks
   have significantly smaller MTU sizes than the frame sizes ATRAC can
   handle, this payload format allows for the fragmentation of an ATRAC
   frame over multiple RTP packets. However, to simplify the
   implementation of RTP receivers, an RTP packet MUST carry either one
   or more complete ATRAC frames or a single fragment of one ATRAC
   frame. In other words, RTP packets MUST NOT contain fragments of
   multiple ATRAC frames and MUST NOT contain a mix of complete and
   fragmented frames.

4.4. Transmission of Redundant Frames

   As RTP does not guarantee reliable transmission, receipt of data is
   not assured. Loss of a packet can result in a "decoding gap" at the
   receiver. One method to remedy this problem is to allow time-shifted
   copies of ATRAC frames to be sent along with current data. For a
   modest cost in latency and implementation complexity, error
   resiliency to packet loss can be achieved. For further details, see
   Section 5.3.2.1 and [12].

4.5. Scalable Lossless Streaming (High-Speed Transfer Mode)

   As ATRAC supports a variation on scalable encoding, this payload
   format provides a mechanism for transmitting essential data (also
   referred to as the base layer) with its enhancement data in two ways
   -- multiplexed through one session or separated over two sessions.

   In either method, only the base layer is essential in producing audio
   data. The enhancement layer carries the remaining audio data needed
   to decode lossless audio data. So in situations of limited
   bandwidth, the sender may choose not to transmit enhancement data yet
   still provide a client with enough data to generate lossily-encoded
   audio through the base layer.

4.5.1. Scalable Multiplexed Streaming

   In multiplexed streaming, the base layer and enhancement layer are
   coupled together in each packet, utilizing only one session as
   illustrated in Figure 1.

   The packet MUST begin with the base layer, and the two layer types
   MUST interleave if both of the layers exist in a packet (only base or
   enhancement is included in a packet at the beginning of a streaming,
   or during the fragmentation).

```
   +----------------+  +----------------+  +----------------+
   |Base|Enhancement|--|Base|Enhancement|--|Base|Enhancement|  ...
   +----------------+  +----------------+  +----------------+
          N                   N+1                 N+2         : Packet
```
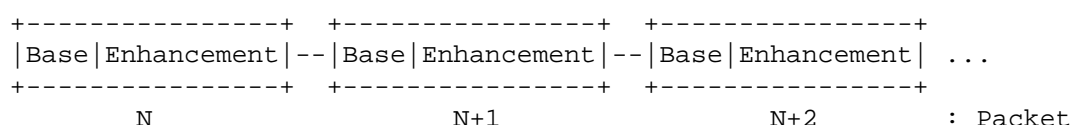
                     Figure 1. Multiplexed Structure

4.5.2. Scalable Multi-Session Streaming

   In multi-session streaming, the base layer and enhancement layer are
   sent over two separate sessions, allowing clients with certain
   bandwidth limitations to receive just the base layer for decoding as
   illustrated in Figure 2.

   In this case, it is REQUIRED to determine which sessions are paired
   together in receiver side. For paired base and enhancement layer
   sessions, the CNAME bindings in the RTP Control Protocol (RTCP)
   session MUST be applied using the same CNAME to ensure correct
   mapping to the RTP source.

While there may be alternative methods for synchronization of the
layers, the timestamp SHOULD be used for synchronizing the base layer
with its enhancement. The two sessions MUST be synchronized using
the information in RTCP SR packets to align the RTP timestamps.

If the enhancement layer's session data cannot arrive until the
presentation time, the decoder MUST decode the base layer session's
data only, ignoring the enhancement layer's data.

```
        Session 1:
        +------+  +------+  +------+  +------+
        | Base |--| Base |--| Base |--| Base | ...
        +------+  +------+  +------+  +------+
           N         N+1       N+2       N+3      : Packet

        Session 2:
        +-------------+  +-------------+  +-------------+
        | Enhancement |--| Enhancement |--| Enhancement | ...
        +-------------+  +-------------+  +-------------+
              N                N+1              N+2         : Packet
```
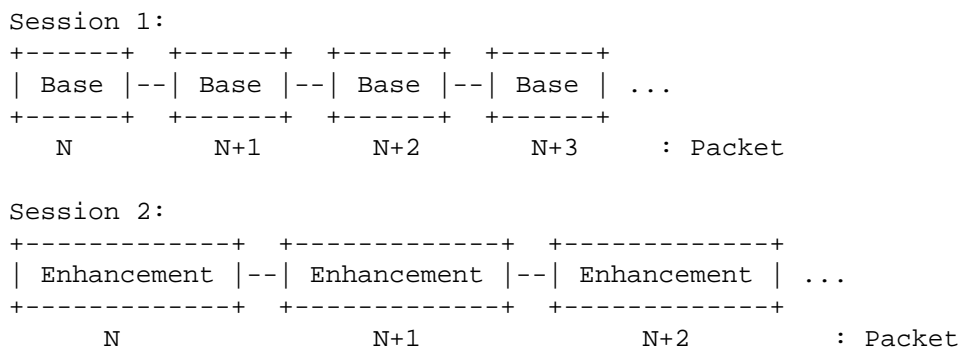
                   Figure 2. Multi-Session Streaming

5. Payload Format

5.1. Global Structure of Payload Format

   The structure of ATRAC Payload is illustrated in Figure 3. The RTP
   payload following the RTP header contains two octet-aligned data
   sections.

```
          +------+-------------+----------------------------+
          |RTP   | ATRAC Header |    ATRAC Frames Section    |
          |Header| Section     | (including redundant data) |
          +------+-------------+----------------------------+
          < --------------- RTP Packet Payload ------------ >
```
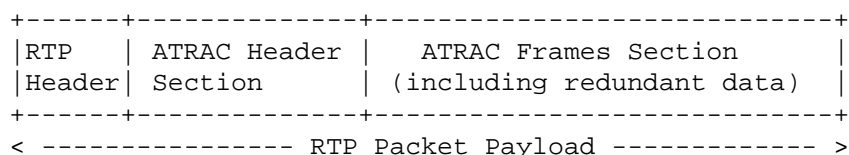
            Figure 3. Structure of RTP Payload of ATRAC Family

   The first data section is the ATRAC Header, containing just one
   header with information for the whole packet. The second section is
   where the encoded ATRAC frames are stored. This may contain either a
   single fragment of one ATRAC frame or one or more complete ATRAC
   frames. The ATRAC Frames Section MUST NOT be empty. When using the
   redundancy mechanism described in Section 5.3.2.1, the redundant
   frame data can be included in this section and timestamp MUST be set
   to the oldest redundant frame's timestamp.

   To benefit from ATRAC's High-Speed Transfer mode lossless encoding
   capability, the RTP payload can be split across two sessions, with
   one transmitting an essential base layer and the other transmitting
   enhancement data. However, in either case, the above structure still
   applies.

5.2. Usage of RTP Header Fields

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            contributing source (CSRC) identifiers             |
|                             .....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
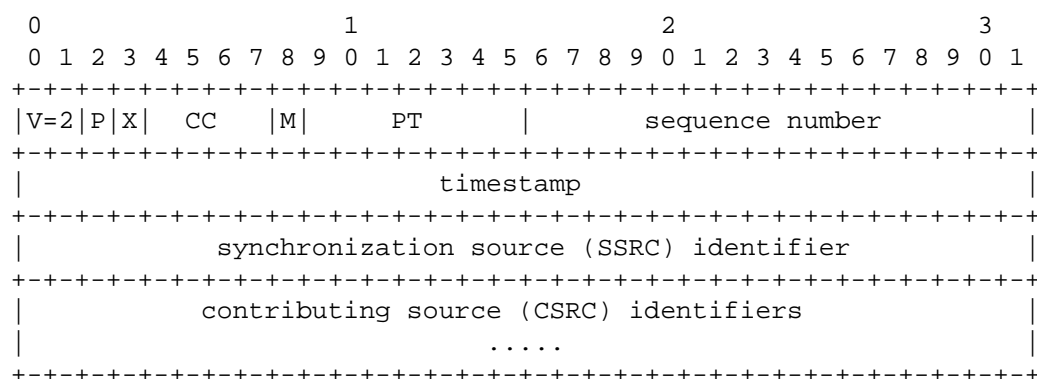
                  Figure 4. RTP Standard Header Part

   The structure of the RTP Standard Header Part is illustrated in
   Figure 4.

   Version(V): 2 bits
   Set to 2.

   Padding(P): 1 bit
   If the padding bit is set, the packet contains one or more additional
   padding octets at the end, which are not part of the payload. The
   last octet of the padding contains a count of how many padding octets
   should be ignored, including itself. Padding may be needed by some
   encryption algorithms with fixed block sizes or for carrying several
   RTP packets in a lower-layer protocol data unit (see [1]).

   Extension(X): 1 bit
   Defined by the RTP profile used.

   CSRC count(CC): 4 bits
   See RFC 3550 [1].

   Marker (M): 1 bit
   Set to 1 if the packet is the first packet after a silence period;
   otherwise, it MUST be set to 0.

Payload Type (PT): 7 bits
The assignment of an RTP payload type for this packet format is
outside the scope of this document; it is specified by the RTP
profile under which this payload format is used, or signaled
dynamically out-of-band (e.g., using the Session Description Protocol
(SDP)).

sequence number: 16 bits
A sequential number for the RTP packet. It ranges from 0 to 65535
and repeats itself periodically.

Timestamp: 32 bits
A timestamp representing the sampling time of the first sample of the
first ATRAC frame in the current RTP packet.
When using SDP, the clock rate of the RTP timestamp MUST be expressed
using the "rtpmap" attribute. For ATRAC3 and ATRAC Advanced
Lossless, the RTP timestamp rate MUST be 44100 Hz. For ATRAC-X, the
RTP timestamp rate is 44100 Hz or 48000 Hz, and it will be selected
by out-of-band signaling.

SSRC: 32 bits
See RFC 3550 [1].

CSRC list: 0 to 15 items, 32 bits each
See RFC 3550 [1].

5.3. RTP Payload Structure

5.3.1. Usage of ATRAC Header Section

The ATRAC header section has the fixed length of one byte as
illustrated in Figure 5.

```
              0 1 2 3 4 5 6 7
             +-+-+-+-+-+-+-+-+
             |C|FrgNo|NFrames|
             +-+-+-+-+-+-+-+-+
```

Figure 5. ATRAC RTP Header

Continuation Flag (C) : 1 bit
The packet that corresponds to the last part of the audio frame data
in a fragmentation MUST have this bit set to 0; otherwise, it's set
to 1.

Fragment Number (FrgNo): 3 bits
In the event of data fragmentation, this value is one for the first
packet, and increases sequentially for the remaining fragmented data

packets. This value MUST be zero for an unfragmented frame. (Note:
3 bits is sufficient to avoid Fragment Number rollover given the
current maximum supported bit-rate in the ATRAC specification. If
that changes, the choice of 3 bits for the Fragment Number should be
revisited.)

Number of Frames (NFrames): 4 bits
The number of audio frames in this packet are field value + 1. This
allows for a maximum of 16 ATRAC-encoded audio frames per packet,
with 0 indicating one audio frame. Each audio frame MUST be complete
in the packet if fragmentation is not applied. In the case of
fragmentation, the data for only one audio frame is allowed to be
fragmented, and this value MUST be 0.

5.3.2. Usage of ATRAC Frames Section

The ATRAC Frames Section contains an integer number of complete ATRAC
frames or a single fragment of one ATRAC frame, as illustrated in
Figure 6. Each ATRAC frame is preceded by a one-bit flag indicating
the layer type and a Block Length field indicating the size in bytes
of the ATRAC frame. If more than one ATRAC frame is present, then
the frames are concatenated into a contiguous string of bit-flag,
Block Length, and ATRAC frame in order of their frame number. This
section MUST NOT be empty.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E|      Block Length       |          ATRAC frame         |...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
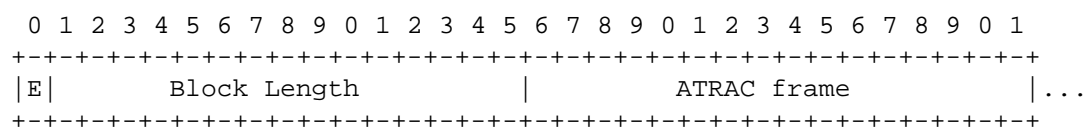
Figure 6. ATRAC Frame Section Format

Layer Type Flag (E): 1 bit
Set to 1 if the corresponding ATRAC frame is from an enhancement
layer. 0 indicates a base layer encoded frame.

Block length: 15 bits
The byte length of encoded audio data for the following frame. This
is so that in the case of fragmentation, if only a subsequent packet
is received, decoding can still occur. 15 bits allows for a maximum
block length of 32,767 bytes.

ATRAC frame: The encoded ATRAC audio data.

5.3.2.1. Support of Redundancy

   This payload format provides a rudimentary scheme to compensate for
   occasional packet loss. As every packet's timestamp corresponds to
   the first audio frame regardless of whether or not it is redundant,
   and because we know how many frames of audio each packet
   encapsulates, if two successive packets are successfully transmitted,
   we can calculate the number of redundant frames being sent. The
   result gives the client a sense of how the server is responding to
   RTCP reports and warns it to expand its buffer size if necessary. As
   an example of using the Redundant Data, refer to Figures 7 and 8.

   In this example, the server has determined that for the next few
   packets, it should send the last two frames from the previous packet
   due to recent RTCP reports. Thus, between packets N and N+1, there
   is a redundancy of two frames (of which the client may choose to
   dispose). The benefit arises when packets N+2 and N+3 do not arrive
   at all, after which eventually packet N+4 arrives with successive
   necessary audio frame data.

   [Sender]

   |-Fr0-|-Fr1-|-Fr2-| Packet: N,   TS=0
         |-Fr1-|-Fr2-|-Fr3-| Packet: N+1, TS=1024
               |-Fr2-|-Fr3-|-Fr4-| Packet: N+2, TS=2048
                     |-Fr3-|-Fr4-|-Fr5-| Packet: N+3, TS=3072
                           |-Fr4-|-Fr5-|-Fr6-| Packet: N+4, TS=4096

   -----------> Packet "N+2" and "N+3" not arrived ------------->

   [Receiver]

   |-Fr0-|-Fr1-|-Fr2-| Packet: N,   TS=0
         |-Fr1-|-Fr2-|-Fr3-| Packet: N+1, TS=1024
                           |-Fr4-|-Fr5-|-Fr6-| Packet: N+4, TS=4096

   The receiver can decode from FR4 to Fr6 by using Packet "N+4" data
   even if the packet loss of "N+2" and "N+3" has occurred.

                      Figure 7. Redundant Example

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         timestamp (= start sample time of Fr1)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            synchronization source (SSRC) identifier           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            contributing source (CSRC) identifiers             |
|                            .....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|  0  |   3   |0|           Block Length      |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          (redundant)  ATRAC frame (Fr1) data  ...             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|       Block Length       |(redundant) ATRAC frame (Fr2)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    (cont.)  |0|   Block Length      |    ATRAC frame (Fr3)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            (cont.)                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
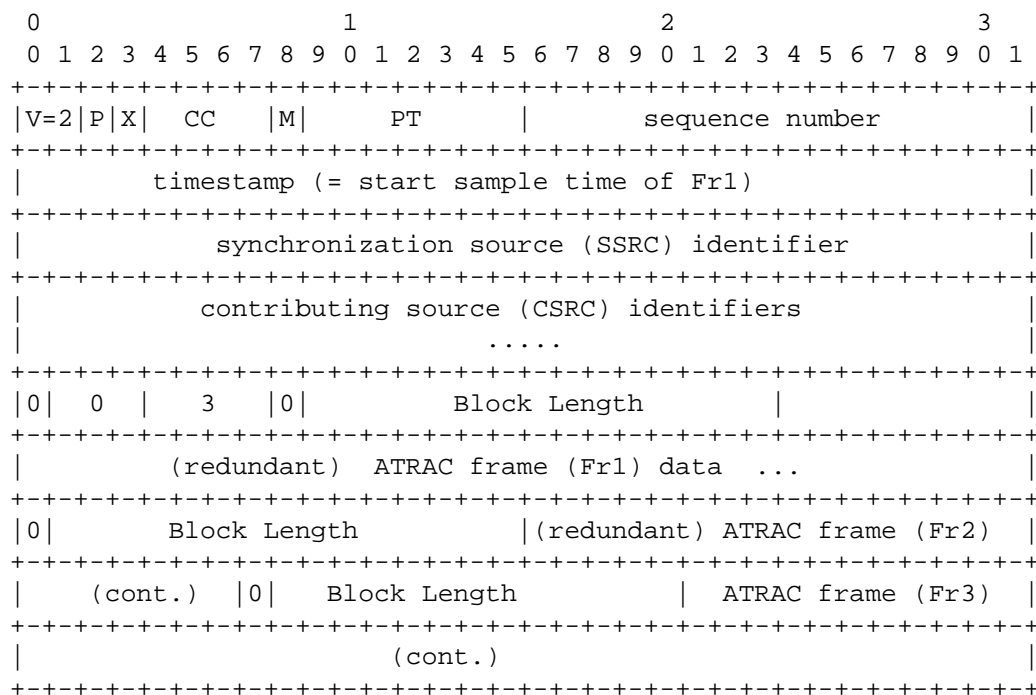
Figure 8. Packet Structure Example with Redundant Data
(Case of Packet "N+1")

5.3.2.2. Frame Fragmentation

Each RTP packet MUST contain either an integer number of ATRAC-
encoded audio frames (with a maximum of 16) or one ATRAC frame
fragment. In the former case, as many complete ATRAC frames as can
fit in a single path-MTU SHOULD be placed in an RTP packet. However,
if even a single ATRAC frame will not fit into a complete RTP packet,
the ATRAC frame MUST be fragmented.

The start of a fragmented frame gets placed in its own RTP packet
with its Continuation bit (C) set to one, and its Fragment Number
(FragNo) set to one. As the frame must be the only one in the
packet, the Number of Frames field is zero. Subsequent packets are
to contain the remaining fragmented frame data, with the Fragment
Number increasing sequentially and the Continuation bit (C)
consistently set to one. As subsequent packets do not contain any
new frames, the Number of Frames field MUST be ignored. The last
packet of fragmented data MUST have the Continuation bit (C) set to
zero.

   Packets containing related fragmented frames MUST have identical
   timestamps. Thus, while the Continuous bit and Fragment Number
   fields indicate fragmentation and a means to reorder the packets, the
   timestamp can be used to determine which packets go together.

6. Packetization Examples

6.1. Example Multi-Frame Packet

   Multiple encoded audio frames are combined into one packet. Note
   how, for this example, only base layer frames are sent redundantly,
   but are followed by interleaved base layer and enhancement layer
   frames as illustrated in Figure 9.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            contributing source (CSRC) identifiers             |
|                             .....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|  0  |  5  |0|          Block Length         |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          (redundant)  base layer frame 1 data...              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|       Block Length         |(redundant) base layer frame 2 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    (cont.)  |0|   Block Length          |  base layer frame 3 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| (cont.) |1|      Block Length           | enhancement frame 3 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| (cont.) |0|      Block Length           |  base layer frame 4 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| (cont.) |1|      Block Length           | enhancement frame 4 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

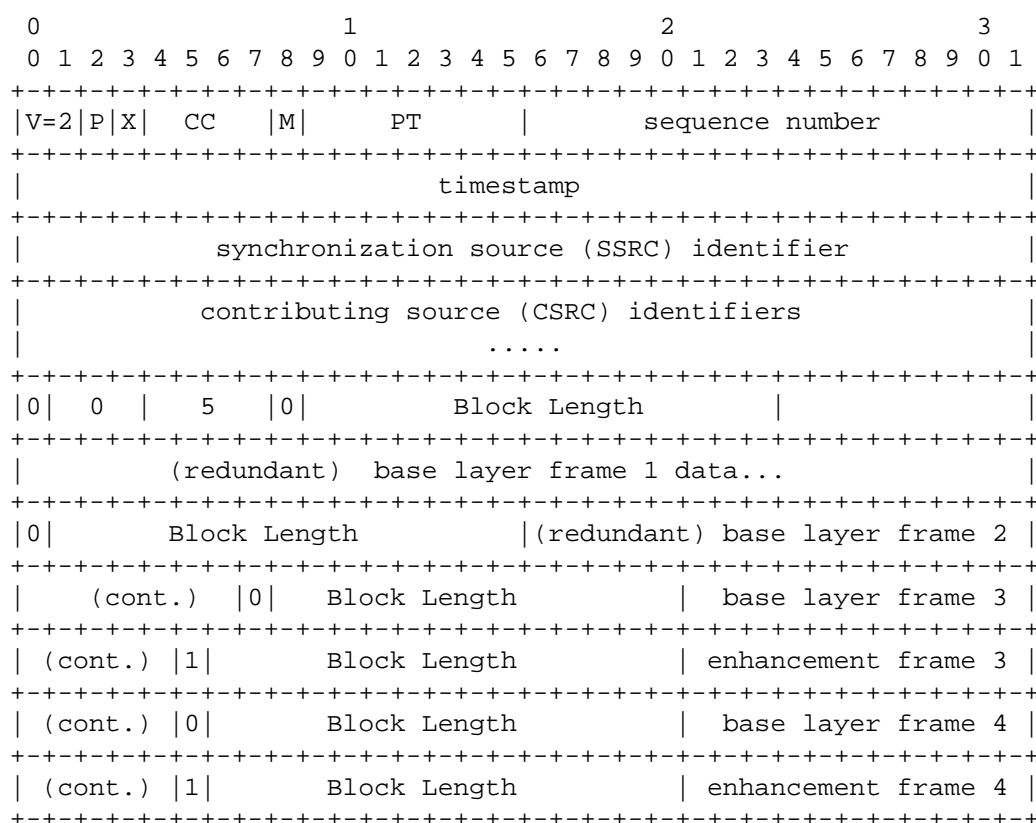   Figure 9. Example Multi-Frame Packet

6.2. Example Fragmented ATRAC Frame

   The encoded audio data frame is split over three RTP packets as
   illustrated in Figure 10. The following points are highlighted in
   the example below:

   o transition from one to zero of the Continuation bit (C)

   o sequential increase in the Fragment Number

   Packet 1:
    0   1   2   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+
   |V=2|P|X| CC |M| PT | sequence number |
   +-+
   | timestamp |
   +-+
   | synchronization source (SSRC) identifier |
   +-+
   | contributing source (CSRC) identifiers |
   | ..... |
   +-+
   |1| 1 | 0 |1| Block Length | |
   +-+
   | enhancement data... |
   +-+

   Packet 2:
    0   1   2   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+
   |V=2|P|X| CC |M| PT | sequence number |
   +-+
   | timestamp |
   +-+
   | synchronization source (SSRC) identifier |
   +-+
   | contributing source (CSRC) identifiers |
   | ..... |
   +-+
   |1| 2 | 0 |1| Block Length | |
   +-+
   | ...more enhancement data... |
   +-+

```
Packet 3:
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |        sequence number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            contributing source (CSRC) identifiers             |
|                             .....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|  3  |  0  |1|          Block Length       |                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                ...the last of the enhancement data            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

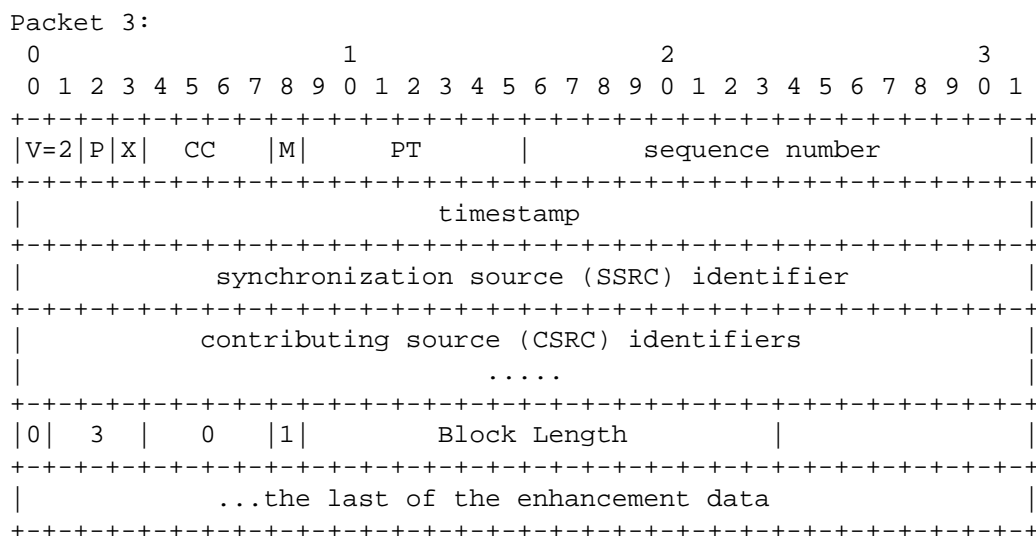Figure 10. Example Fragmented ATRAC Frame

7. Payload Format Parameters

   Certain parameters will need to be defined before ATRAC-family-
   encoded content can be streamed. Other optional parameters may also
   be defined to take advantage of specific features relevant to certain
   ATRAC versions. Parameters for ATRAC3, ATRAC-X, and ATRAC Advanced
   Lossless are defined here as part of the media subtype registration
   process. A mapping of these parameters into the Session Description
   Protocol (SDP) (RFC 4566) [2] is also provided for applications that
   utilize SDP. These registrations use the template defined in RFC
   4288 [5] and follow RFC 4855 [6].

   The data format and parameters are specified for real-time transport
   in RTP.

7.1. ATRAC3 Media Type Registration

   The media subtype for the Adaptive TRansform Codec version 3 (ATRAC3)
   uses the template defined in RFC 4855 [6].

   Note, any unknown parameter MUST be ignored by the receiver.

   Type name: audio

   Subtype name: ATRAC3

Required parameters:

rate: Represents the sampling frequency in Hz of the original audio
data. Permissible value is 44100 only.

baseLayer: Indicates the encoded bit-rate in kbps for the audio data
to be streamed. Permissible values are 66, 105, and 132.

Optional parameters:

ptime: See RFC 4566 [2].

maxptime: See RFC 4566 [2].
The frame length of ATRAC3 is 1024/44100 = 23.22...(ms), and
fractional value may not be applicable for the SDP definition.

So the value of the parameter MUST be a multiple of 24 (ms)
considering safe transmission.

If this parameter is not present, the sender MAY encapsulate a
maximum of 6 encoded frames into one RTP packet, in streaming of
ATRAC3.

maxRedundantFrames: The maximum number of redundant frames that may
be sent during a session in any given packet under the redundant
framing mechanism detailed in the document. Allowed values are
integers in the range of 0 to 15, inclusive. If this parameter is
not used, a default of 15 MUST be assumed.

Encoding considerations: This media type is framed and contains
binary data.

Security considerations: This media type does not carry active
content. See Section 9 of this document.

Interoperability considerations: none

Published specification: ATRAC3 Standard Specification [9]

Applications that use this media type:
Audio and video streaming and conferencing tools.

Additional information: none
Magic number(s): none
File extension(s): 'at3', 'aa3', and 'omg'
Macintosh file type code(s): none

   Person and email address to contact for further information:
   Mitsuyuki Hatanaka
   Jun Matsumoto
   actech@jp.sony.com

   Intended usage: COMMON

   Restrictions on usage: This media type depends on RTP framing, and
   hence is only defined for transfer via RTP.

   Author:
   Mitsuyuki Hatanaka
   Jun Matsumoto
   actech@jp.sony.com

   Change controller: IETF AVT WG delegated from the IESG

7.2. ATRAC-X Media Type Registration

   The media subtype for the Adaptive TRansform Codec version X
   (ATRAC-X) uses the template defined in RFC 4855 [6].

   Note, any unknown parameter MUST be ignored by the receiver.

   Type name: audio

   Subtype name: ATRAC-X

   Required parameters:

   rate: Represents the sampling frequency in Hz of the original
   audio data. Permissible values are 44100 and 48000.

   baseLayer: Indicates the encoded bit-rate in kbps for the audio data
   to be streamed. Permissible values are 32, 48, 64, 96, 128, 160,
   192, 256, 320, and 352.

   channelID: Indicates the number of channels and channel layout
   according to the table1 in Section 7.4. Note that this layout is
   different from that proposed in RFC 3551 [3]. However, as channelID
   = 0 defines an ambiguous channel layout, the channel mapping defined
   in Section 4.1 of [3] could be used. Permissible values are 0, 1, 2,
   3, 4, 5, 6, 7.

   Optional parameters:

   ptime: See RFC 4566 [2].

maxptime: See RFC 4566 [2].
The frame length of ATRAC-X is 2048/44100 = 46.44...(ms) or
2048/48000 = 42.67...(ms), but fractional value may not be applicable
for the SDP definition. So the value of the parameter MUST be a
multiple of 47 (ms) or 43 (ms) considering safe transmission.

If this parameter is not present, the sender MAY encapsulate a
maximum of 16 encoded frames into one RTP packet, in streaming of
ATRAC-X.

maxRedundantFrames: The maximum number of redundant frames that may
be sent during a session in any given packet under the redundant
framing mechanism detailed in the document. Allowed values are
integers in the range 0 to 15, inclusive. If this parameter is not
used, a default of 15 MUST be assumed.

delayMode: Indicates a desire to use low-delay features, in which
case the decoder will process received data accordingly based on this
value. Permissible values are 2 and 4.

Encoding considerations: This media type is framed and contains
binary data.

Security considerations: This media type does not carry active
content. See Section 9 of this document.

Interoperability considerations: none

Published specification: ATRAC-X Standard Specification [10]

Applications that use this media type:
Audio and video streaming and conferencing tools.

Additional information: none

Magic number(s): none
File extension(s): 'atx', 'aa3', and 'omg'
Macintosh file type code(s): none

Person and email address to contact for further information:
Mitsuyuki Hatanaka
Jun Matsumoto
actech@jp.sony.com

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and
hence is only defined for transfer via RTP.

   Author:
   Mitsuyuki Hatanaka
   Jun Matsumoto
   actech@jp.sony.com

   Change controller: IETF AVT WG delegated from the IESG

7.3. ATRAC Advanced Lossless Media Type Registration

   The media subtype for the Adaptive TRansform Codec Lossless version
   (ATRAC Advanced Lossless) uses the template defined in RFC 4855 [6].

   Note, any unknown parameter MUST be ignored by the receiver.

   Type name: audio

   Subtype name: ATRAC-ADVANCED-LOSSLESS

   Required parameters:

   rate: Represents the sampling frequency in Hz of the original
   audio data. Permissible value is 44100 only for High-Speed Transfer
   mode. Any value of 24000, 32000, 44100, 48000, 64000, 88200, 96000,
   176400, and 192000 can be used for Standard mode.

   baseLayer: Indicates the encoded bit-rate in kbps for the base layer
   in High-Speed Transfer mode lossless encodings.

   For Standard lossless mode, this value MUST be 0.

   The Permissible values for ATRAC3 baselayer are 66, 105, and 132.
   For ATRAC-X baselayer, they are 32, 48, 64, 96, 128, 160, 192, 256,
   320, and 352.

   blockLength: Indicates the block length. In High-Speed Transfer
   mode, the value of 1024 and 2048 is used for ATRAC3 based and ATRAC-X
   based ATRAC Advanced Lossless streaming, respectively.

   Any value of 512, 1024, and 2048 can be used for Standard mode.

   channelID: Indicates the number of channels and channel layout
   according to the table1 in Section 7.4. Note that this layout is
   different from that proposed in RFC 3551 [3]. However, as channelID
   = 0 defines an ambiguous channel layout, the channel mapping defined
   in Section 4.1 of [3] could be used in this case. Permissible values
   are 0, 1, 2, 3, 4, 5, 6, 7.

   ptime: See RFC 4566 [2].

   maxptime: See RFC 4566 [2].
   In streaming of ATRAC Advanced Lossless, multiple frames cannot be
   transmitted in a single RTP packet, as the frame size is large. So
   it SHOULD be regarded as the time of one encoded frame in both of the
   sender and the receiver side. The frame length of ATRAC Advanced
   Lossless is 512/44100 = 11.6...(ms), 1024/44100 = 23.22...(ms), or
   2048/44100 = 46.44...(ms), but fractional value may not be applicable
   for the SDP definition. So the value of the parameter MUST be
   12(ms), 24(ms), or 47(ms) considering safe transmission.

   Encoding considerations: This media type is framed and contains
   binary data.

   Security considerations: This media type does not carry active
   content. See Section 9 of this document.

   Interoperability considerations: none

   Published specification:
   ATRAC Advanced Lossless Standard Specification [11]

   Applications that use this media type:
   Audio and video streaming and conferencing tools.

   Additional information: none

   Magic number(s): none
   File extension(s): 'aal', 'aa3', and 'omg'
   Macintosh file type code(s): none

   Person and email address to contact for further information:

   Mitsuyuki Hatanaka
   Jun Matsumoto
   actech@jp.sony.com

   Intended usage: COMMON

   Restrictions on usage: This media type depends on RTP framing, and
   hence is only defined for transfer via RTP.

   Author:
   Mitsuyuki Hatanaka
   Jun Matsumoto
   actech@jp.sony.com

   Change controller: IETF AVT WG delegated from the IESG

7.4. Channel Mapping Configuration Table

   Table 1 explains the mapping between the channelID as passed during
   SDP negotiations, and the speaker mapping the value represents.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| channelID | Number of |    Default Speaker     |
|           | Channels  |       Mapping          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     0     |  max 64   |     undefined          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     1     |     1     | front: center          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     2     |     2     | front: left, right     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     3     |     3     | front: left, right     |
|           |           | front: center          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     4     |     4     | front: left, right     |
|           |           | front: center          |
|           |           | rear: surround         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     5     |    5+1     | front: left, right     |
|           |           | front: center          |
|           |           | rear: left, right      |
|           |           | LFE                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     6     |    6+1     | front: left, right     |
|           |           | front: center          |
|           |           | rear: left, right      |
|           |           | rear: center           |
|           |           | LFE                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     7     |    7+1     | front: left, right     |
|           |           | front: center          |
|           |           | rear: left, right      |
|           |           | side: left, right      |
|           |           | LFE                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                  Table 1. Channel Configuration

7.5. Mapping Media Type Parameters into SDP

   The information carried in the Media type specification has a
   specific mapping to fields in the Session Description Protocol (SDP)
   [2], which is commonly used to describe RTP sessions. When SDP is
   used to specify sessions employing the ATRAC family of codecs, the
   following mapping rules according to the ATRAC codec apply.

7.5.1. For Media Subtype ATRAC3

   o The Media type ("audio") goes in SDP "m=" as the media name.

   o The Media subtype (payload format name) goes in SDP "a=rtpmap" as
      the encoding name. ATRAC3 supports only mono or stereo signals,
      so a corresponding number of channels (0 or 1) MUST also be
      specified in this attribute.

   o The "baseLayer" parameter goes in SDP "a=fmtp". This parameter
      MUST be present. "maxRedundantFrames" may follow, but if no value
      is transmitted, the receiver SHOULD assume a default value of
      "15".

   o The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and
      "a=maxptime" attributes, respectively.

7.5.2. For Media Subtype ATRAC-X

   o The Media type ("audio") goes in SDP "m=" as the media name.

   o The Media subtype (payload format name) goes in SDP "a=rtpmap" as
      the encoding name. This SHOULD be followed by the "sampleRate"
      (as the RTP clock rate), and then the actual number of channels
      regardless of the channelID parameter.

   o The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and
      "a=maxptime" attributes, respectively.

   o Any remaining parameters go in the SDP "a=fmtp" attribute by
      copying them directly from the Media type string as a semicolon-
      separated list of parameter=value pairs. The "baseLayer"
      parameter MUST be the first entry on this line. The "channelID"
      parameter MUST be the next entry. The receiver MUST assume a
      default value of "15" for "maxRedundantFrames".

7.5.3. For Media Subtype ATRAC Advanced Lossless

   o The Media type ("audio") goes in SDP "m=" as the media name.

   o The Media subtype (payload format name) goes in SDP "a=rtpmap" as
      the encoding name. This MUST be followed by the "sampleRate" (as
      the RTP clock rate), and then the actual number of channels
      regardless of the channelID parameter.

   o The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and
      "a=maxptime" attributes, respectively.

   o Any remaining parameters go in the SDP "a=fmtp" attribute by
      copying them directly from the Media type string as a semicolon-
      separated list of parameter=value pairs.

      On this line, the parameters "baseLayer" and "blockLength" MUST be
      present in this order.

      The value of "blockLength" MUST be one of 1024 and 2048, for using
      ATRAC3 and ATRAC-X as baselayer, respectively. If "baseLayer=0"
      (means standard mode), "blockLength" MUST be one of either 512,
      1024, or 2048. The "channelID" parameter MUST be the next entry .
      The receiver MUST assume a default value of "15" for
      "maxRedundantFrames".

7.6. Offer/Answer Model Considerations

   Some options for encoding and decoding ATRAC audio data will require
   either or both of the sender and receiver complying with certain
   specifications. In order to establish an interoperable transmission
   framework, an Offer/Answer negotiation in SDP MUST observe the
   following considerations. (See [14].)

7.6.1. For All Three Media Subtypes

   o Each combination of the RTP payload transport format configuration
      parameters (baseLayer and blockLength, sampleRate, channelID) is
      unique in its bit-pattern and not compatible with any other
      combination. When creating an offer in an application desiring to
      use the more advanced features (sample rates above 44100 kHz, more
      than two channels), the offerer SHOULD also offer a payload type
      containing only the lowest set of necessary requirements. If
      multiple configurations are of interest to the application, they
      may all be offered.

   o The parameters "maxptime" and "ptime" will in most cases not
      affect interoperability; however, the setting of the parameters
      can affect the performance of the application. The SDP
      Offer/Answer handling of the "ptime" parameter is described in RFC
      3264. The "maxptime" parameter MUST be handled in the same way.

7.6.2. For Media Subtype ATRAC3

   o In response to an offer, downgraded subsets of "baseLayer" are
      possible. However, for best performance, we suggest the answer
      contain the highest possible values offered.

7.6.3. For Media Subtype ATRAC-X

   o In response to an offer, downgraded subsets of "sampleRate",
      "baseLayer", and "channelID" are possible. For best performance,
      an answer MUST NOT contain any values requiring further
      capabilities than the offer contains, but it SHOULD provide values
      as close as possible to those in the offer.

   o The "maxRedundantFrames" is a suggested minimum. This value MAY
      be increased in an answer (with a maximum of 15), but MUST NOT be
      reduced.

   o The optional parameter "delayMode" is non-negotiable. If the
      Answerer cannot comply with the offered value, the session MUST be
      deemed inoperable.

7.6.4. For Media Subtype ATRAC Advanced Lossless

   o In response to an offer, downgraded subsets of "sampleRate",
      "baseLayer", and "channelID" are possible. For best performance,
      an answer MUST NOT contain any values requiring further
      capabilities than the offer contains, but it SHOULD provide values
      as close as possible to those in the offer.

   o There are no requirements when negotiating "blockLength", other
      than that both parties must be in agreement.

   o The "maxRedundantFrames" is a suggested minimum. This value MAY
      be increased in an answer (with a maximum of 15), but MUST NOT be
      reduced.

   o For transmission of scalable multi-session streaming of ATRAC
      Advanced Lossless content, the attributes of media stream
      identification, group information, and decoding dependency between
      base layer stream and enhancement layer stream MUST be signaled in
      SDP by the Offer/Answer model. In this case, the attribute of

      "group", "mid", and "depend" followed by the appropriate parameter
      MUST be used in SDP [7] [8] in order to indicate layered coding
      dependency. The attribute of "group" followed by "DDP" parameter
      is used for indicating the relationship between the base and the
      enhancement layer stream with decoding dependency. Each stream is
      identified by "mid" attribute, and the dependency of enhancement
      layer stream is defined by the "depend" attribute, as the
      enhancement layer is only useful when the base layer is available.
      Examples for signaling ATRAC Advanced Lossless decoding dependency
      are described in Sections 7.8 and 7.9.

7.7. Usage of Declarative SDP

   In declarative usage, like SDP in Real-Time Streaming Protocol (RTSP)
   [15] or Session Announcement Protocol (SAP) [16], the parameters MUST
   be interpreted as follows:

   o The payload format configuration parameters (baseLayer,
      sampleRate, channelID) are all declarative and a participant MUST
      use the configuration(s) provided for the session. More than one
      configuration may be provided if necessary by declaring multiple
      RTP payload types; however, the number of types SHOULD be kept
      small.

   o Any "maxptime" and "ptime" values SHOULD be selected with care to
      ensure that the session's participants can achieve reasonable
      performance.

   o The attribute of "mid", "group", and "depend" MUST be used for
      indicating the relationship and dependency of the base layer and
      the enhancement layer in scalable multi-session streaming of ATRAC
      ADVANCED LOSSLESS content, as described in Sections 7.6, 7.8, and
      7.9.

7.8. Example SDP Session Descriptions

   Example usage of ATRAC-X with stereo at 44100 Hz:

   v=0
   o=atrac 2465317890 2465317890 IN IP4 service.example.com
   s=ATRAC-X Streaming
   c=IN IP4 192.0.2.1/127
   t=3409539540 3409543140
   m=audio 49120 RTP/AVP 99
   a=rtpmap:99 ATRAC-X/44100/2
   a=fmtp:99 baseLayer=128; channelID=2; delayMode=2
   a=maxptime:47

Example usage of ATRAC-X with 5.1 setup at 48000 Hz:

```
v=0
o=atrac 2465317890 2465317890 IN IP4 service.example.com
s=ATRAC-X 5.1ch Streaming
c=IN IP4 192.0.2.1/127
t=3409539540 3409543140
m=audio 49120 RTP/AVP 99
a=rtpmap:99 ATRAC-X/48000/6
a=fmtp:99 baseLayer=320; channelID=5
a=maxptime:43
```

Example usage of ATRAC-Advanced-Lossless in multiplexed
High-Speed Transfer mode:

```
v=0
o=atrac 2465317890 2465317890 IN IP4 service.example.com
s=AAL Multiplexed Streaming
c=IN IP4 192.0.2.1/127
t=3409539540 3409543140
m=audio 49200 RTP/AVP 96
a=rtpmap:96 ATRAC-ADVANCED-LOSSLESS/44100/2
a=fmtp:96 baseLayer=128; blockLength=2048; channelID=2
a=maxptime:47
```

Example usage of ATRAC-Advanced-Lossless in multi-session High-Speed
Transfer mode. In this case, the base layer and the enhancement
layer stream are identified by L1 and L2, respectively, and L2
depends on L1 in decoding.

```
v=0
o=atrac 2465317890 2465317890 IN IP4 service.example.com
s=AAL Multi Session Streaming
c=IN IP4 192.0.2.1/127
t=3409539540 3409543140
a=group:DDP L1 L2
m=audio 49200 RTP/AVP 96
a=rtpmap:96 ATRAC-ADVANCED-LOSSLESS/44100/2
a=fmtp:96 baseLayer=128; blockLength=2048; channelID=2
a=maxptime:47
a=mid:L1
m=audio 49202 RTP/AVP 97
a=rtpmap:97 ATRAC-ADVANCED-LOSSLESS/44100/2
a=fmtp:97 baseLayer=0; blockLength=2048; channelID=2
a=maxptime:47
a=mid:L2
a=depend:97 lay L1:96
```

   Example usage of ATRAC-Advanced-Lossless in Standard mode:

   m=audio 49200 RTP/AVP 99
   a=rtpmap:99 ATRAC-ADVANCED-LOSSLESS/44100/2
   a=fmtp:99 baseLayer=0; blockLength=1024; channelID=2
   a=maxptime:24

7.9. Example Offer/Answer Exchange

   The following Offer/Answer example shows how a desire to stream
   multi-channel content is turned down by the receiver, who answers
   with only the ability to receive stereo content:

   Offer:

   m=audio 49170 RTP/AVP 98 99
   a=rtpmap:98 ATRAC-X/44100/6
   a=fmtp:98 baseLayer=320; channelID=5
   a=rtpmap:99 ATRAC-X/44100/2
   a=fmtp:99 baseLayer=160; channelID=2

   Answer:

   m=audio 49170 RTP/AVP 99
   a=rtpmap:99 ATRAC-X/44100/2
   a=fmtp:99 baseLayer=160; channelID=2

   The following Offer/Answer example shows the receiver answering with
   a selection of supported parameters:

   Offer:

   m=audio 49170 RTP/AVP 97 98 99
   a=rtpmap:97 ATRAC-X/44100/2
   a=fmtp:97 baseLayer=128; channelID=2
   a=rtpmap:98 ATRAC-X/44100/6
   a=fmtp:98 baseLayer=128; channelID=5
   a=rtpmap:99 ATRAC-X/48000/6
   a=fmtp:99 baseLayer=320; channelID=5

   Answer:

   m=audio 49170 RTP/AVP 97 98
   a=rtpmap:97 ATRAC-X/44100/2
   a=fmtp:97 baseLayer=128; channelID=2
   a=rtpmap:98 ATRAC-X/44100/6
   a=fmtp:98 baseLayer=128; channelID=5

The following Offer/Answer example shows an exchange in trying to
resolve using ATRAC-Advanced-Lossless. The offer contains three
options: multi-session High-Speed Transfer mode, multiplexed High-
Speed Transfer mode, and Standard mode.

Offer:

// Multi-session High-Speed Transfer mode, L1 and L2 correspond
   to the base layer and the enhancement layer, respectively, and L2
   depends on L1 in decoding.

   a=group:DDP L1 L2
   m=audio 49200 RTP/AVP 96
   a=rtpmap:96 ATRAC-ADVANCED-LOSSLESS/44100/2
   a=fmtp:96 baseLayer=132; blockLength=1024; channelID=2
   a=maxptime:24
   a=mid:L1

   m=audio 49202 RTP/AVP 97
   a=rtpmap:97 ATRAC-ADVANCED-LOSSLESS/44100/2
   a=fmtp:97 baseLayer=0; blockLength=2048; channelID=2
   a=maxptime:24
   a=mid:L2
   a=depend:97 lay L1:96

// Multiplexed High-Speed Transfer mode
   m=audio 49200 RTP/AVP 98
   a=rtpmap:98 ATRAC-ADVANCED-LOSSLESS/44100/2
   a=fmtp:98 baseLayer=256; blockLength=2048; channelID=2
   a=maxptime:47

// Standard mode
   m=audio 49200 RTP/AVP 99
   a=rtpmap:99 ATRAC-ADVANCED-LOSSLESS/44100/2
   a=fmtp:99 baseLayer=0; blockLength=2048; channelID=2
   a=maxptime:47

   Answer:

   a=group:DDP L1 L2
   m=audio 49200 RTP/AVP 94
   a=rtpmap:94 ATRAC-ADVANCED-LOSSLESS/44100/2
   a=fmtp:94 baseLayer=132; blockLength=1024; channelID=2
   a=maxptime:24
   a=mid:L1

```
m=audio 49202 RTP/AVP 95
a=rtpmap:95 ATRAC-ADVANCED-LOSSLESS/44100/2
a=fmtp:95 baseLayer=0; blockLength=2048; channelID=2
a=maxptime:24
a=mid:L2
a=depend:95 lay L1:94
```

Note that the names of payload format (encoding) and Media subtypes
are case-insensitive in both places. Similarly, parameter names are
case-insensitive both in Media types and in the default mapping to
the SDP a=fmtp attribute.

8. IANA Considerations

   Three new Media subtypes, audio/ATRAC3, audio/ATRAC-X, and
   audio/ATRAC-ADVANCED-LOSSLESS, have been registered (see Section 7).

9. Security Considerations

   The payload format as described in this document is subject to the
   security considerations defined in RFC 3550 [1] and any applicable
   profile, for example, RFC 3551 [3]. Also, the security of Media type
   registration MUST be taken into account as described in Section 5 of
   RFC 4855 [6].

   The payload for ATRAC family consists solely of compressed audio data
   to be decoded and presented as sound, and the standard specifications
   of ATRAC3, ATRAC-X, and ATRAC Advanced Lossless [9] [10] [11]
   strictly define the bit stream syntax and the buffer model in decoder
   side for each codec. So they can not carry "active content" that
   could impose malicious side effects upon the receiver, and they do
   not cause any problem of illegal resource consumption in receiver
   side, as far as the bit streams are conforming to their standard
   specifications.

   This payload format does not implement any security mechanisms of its
   own. Confidentiality, integrity protection, and authentication have
   to be provided by a mechanism external to this payload format, e.g.,
   SRTP RFC 3711 [13].

10. Considerations on Correct Decoding

10.1. Verification of the Packets

   Verification of the received encoded audio packets MUST be performed
   so as to ensure correct decoding of the packets. As a most primitive
   implementation, the comparison of the packet size and payload length
   can be taken into account. If the UDP packet length is longer than

the RTP packet length, the packet can be accepted, but the extra
bytes MUST be ignored. In case of receiving a shorter UDP packet or
improperly encoded packets, the packets MUST be discarded.

10.2. Validity Checking of the Packets

Also, validity checking of the received audio packets MUST be
performed. It can be carried out by the decoding process, as the
ATRAC format is designed so that the validity of data frames can be
determined by decoding the algorithm. The required decoder response
to a malformed frame is to discard the malformed data and conceal the
errors in the audio output until a valid frame is detected and
decoded. This is expected to prevent crashes and other abnormal
decoder behavior in response to errors or attacks.

11. References

11.1. Normative References

   [1]   Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson,
          "RTP: A Transport Protocol for Real-Time Applications", STD 64,
          RFC 3550, July 2003.

   [2]   Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
          Description Protocol", RFC 4566, July 2006.

   [3]   Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video
          Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

   [4]   Bradner, S., "Key words for use in RFCs to Indicate Requirement
          Levels", BCP 14, RFC 2119, March 1997.

   [5]   Freed, N. and J. Klensin, "Media Type Specifications and
          Registration Procedures", BCP 13, RFC 4288, December 2005.

   [6]   Casner, S., "Media Type Registration of RTP Payload Formats",
          RFC 4855, February 2007.

   [7]   Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne,
          "Grouping of Media Lines in the Session Description Protocol
          (SDP)", RFC 3388, December 2002.

   [8]   Schierl, T., and S. Wenger, "Signaling Media Decoding
          Dependency in the Session Description Protocol (SDP)", RFC
          5583, July 2009.

   [9]   ATRAC3 Standard Specification ver.1.1, Sony Corporation, 2003.

[10] ATRAC-X Standard Specification ver.1.2, Sony Corporation, 2004.

   [11] ATRAC Advanced Lossless Standard Specification ver.1.1, Sony
         Corporation, 2007.

11.2. Informative References

[12] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley,
         M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP
         Payload for Redundant Audio Data", RFC 2198, September 1997.

   [13] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K.
         Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC
         3711, March 2004.

   [14] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with
         Session Description Protocol (SDP)", RFC 3264, June 2002.

   [15] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming
         Protocol (RTSP)", RFC 2326, April 1998.

   [16] Handley, M., Perkins, C., and E. Whelan, "Session Announcement
         Protocol", RFC 2974, October 2000.

Authors' Addresses

   Mitsuyuki Hatanaka
   Sony Corporation, Japan
   1-7-1 Konan
   Minato-ku
   Tokyo 108-0075
   Japan

   EMail: actech@jp.sony.com

   Jun Matsumoto
   Sony Corporation, Japan
   1-7-1 Konan
   Minato-ku
   Tokyo 108-0075
   Japan

   EMail: actech@jp.sony.com