TECHNICAL REPORT

ISO/IEC TR 14496-24

First edition
2008-01-15

Information technology — Coding of audio-visual objects —

Part 24: Audio and systems interaction

Technologies de l'information — Codage d'objets audiovisuels —

Partie 24: Codage audio et interaction de systèmes

Reference number
ISO/IEC TR 14496-24:2008(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword ... iv

1 Scope ... 1

2 Motivating audio composition time stamp handling ... 1

3 AAC Encoder/Decoder Behavior ... 2
3.1 Example 1: AAC ... 2
3.1.1 Overview ... 2
3.1.2 Pre-roll ... 4
3.1.3 Edit-list ... 4
3.1.4 Compressed Information and Decoder behavior ... 4
3.2 Example 2: HE-AAC ... 5
3.2.1 Overview ... 5

4 Streaming Considerations ... 6

Annex A (informative) Relevant ISO Base Media File Format Syntax ... 7
A.1 Pre-roll syntax ... 7
A.2 Edit-list syntax ... 8

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 14496-24, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC TR 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

— *Part 1: Systems*

— *Part 2: Visual*

— *Part 3: Audio*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Delivery Multimedia Integration Framework (DMIF)*

— *Part 7: Optimized reference software for coding of audio-visual objects*

— *Part 8: Carriage of ISO/IEC 14496 contents over IP networks*

- *Part 9: Reference hardware description*
- *Part 10: Advanced Video Coding*
- *Part 11: Scene description and application engine*
- *Part 12: ISO base media file format*
- *Part 13: Intellectual Property Management and Protection (IPMP) extensions*
- *Part 14: MP4 file format*
- *Part 15: Advanced Video Coding (AVC) file format*
- *Part 16: Animation Framework eXtension (AFX)*
- *Part 17: Streaming text format*
- *Part 18: Font compression and streaming*
- *Part 19: Synthesized texture stream*
- *Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)*
- *Part 21: MPEG-J Graphics Framework eXtensions (GFX)*
- *Part 22: Open Font Format*
- *Part 23: Symbolic Music Representation*
- *Part 24: Audio and systems interaction*

Information technology — Coding of audio-visual objects —

Part 24: Audio and systems interaction

1 Scope

This part of ISO/IEC TR 14496 describes the desired joint behavior of MPEG-4 Systems (MPEG-4 File Format) and MPEG-4 Audio codecs. It is desired that MPEG-4 Audio encoders and decoders permit finite length signals to be encoded to a file (particularly MPEG-4 files) and decoded again to obtain the identical signal, subject to codec distortions. This will allow the use of audio in systems implementations (particularly MPEG-4 Systems), perhaps with other media such as video, in a deterministic fashion. Most importantly, the decoded signal will have nothing “extra” at the beginning or “missing” at the end.

This permits:

a) an exact ‘round trip’ from raw audio to encoded file back to raw audio (excepting encoding artifacts);

b) predictable synchronization between audio and other media such as video;

c) correct behavior when performing random access as well as when starting at the beginning of a stream;

d) identical behavior when edits are applied in the raw domain and the encoded domain (again, excepting encoding artifacts).

It is also required that there be predictable interoperability between encoders (as represented by files) and decoders. There are two kinds of audio ‘offsets’ (or ‘delay’ in the context of transmission): those that result from the encoding process, and those that result from the decoding process. This document is primarily concerned with the latter.

These issues are resolved by the following:

- The handling of composition time stamps for audio composition units is specified. Special care is taken in the case of compressed data, like HE-AAC coded audio, that can be decoded in a backward compatible fashion as well as in an enhanced fashion.

- Examples are given that show how finite length signals can be encoded to an MPEG-4 file and decoded again to obtain the identical signal, excepting codec distortions. Most importantly, the decoded signal has nothing “extra” at the beginning or “missing” at the end.

2 Motivating audio composition time stamp handling

For compressed data, like HE-AAC coded audio, which can be decoded by different decoder configurations, special attention is needed. In this case, decoding can be done in a backward-compatible fashion (AAC only) as well as in an enhanced fashion (AAC+SBR). In order to insure that timestamps are correct (so that audio remains synchronized with other media), the following must considered concerning MPEG-4 Systems and Audio:

- If compressed data permits both backward-compatible and enhanced decoding, and if the decoder is operating in a backwards-compatible fashion, then the decoder does not have to take any action. However if the decoder is operating in enhanced fashion such that it is using a post-processor that inserts some additional delay (e.g., the SBR post-processor in HE-AAC), then it must notify Systems about the additional time delay incurred relative to the backwards-compatible mode. With the delay thus indicated, Systems can handle the timestamps of the composition units as needed so as to compensate for the additional delay.

- Specifically for HE-AAC (using any of the available signaling mechanisms, i.e., implicit signaling, backward compatible explicit signaling, or hierarchical explicit signaling) the original access unit timestamps apply to backward-compatible AAC decoding and timestamp adjustment for delay-compensation is needed in case of AAC+SBR decoding.

Figure 1 shows the composition unit that is generated by an AAC decoder (upper half) and by an HE-AAC decoder operating SBR in dual-rate mode (lower half) when being fed with an access unit of an HE-AAC bitstream that employs backward compatible signaling. Note that the composition time stamp associated to said access unit applies to the *n*-th sample of the composition unit. For the AAC decoder case, *n* has the value 1. For the HE-AAC decoder case, *n* has the value 962+1 to reflect the additional algorithmic delay of 962 samples of the SBR tool at the HE-AAC output sampling rate (which is twice the sampling rate of the backward compatible AAC output).

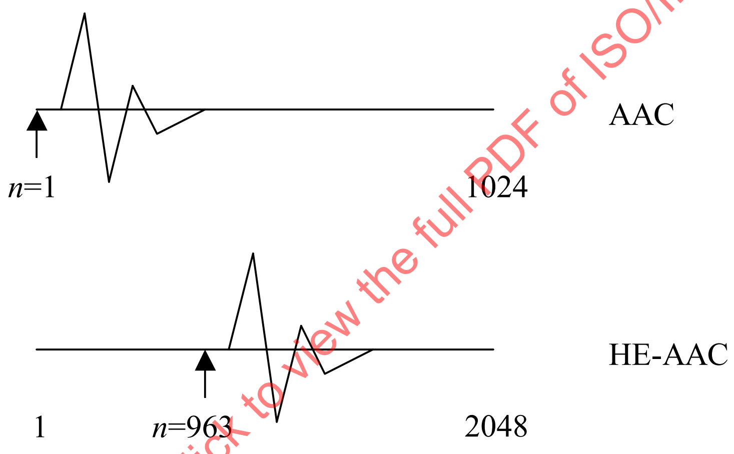

Figure 1 — Composition unit (audio waveform segment) generated by AAC decoder and HE-AAC decoder fed with the same access unit (bitstream frame)

The timestamp handling depends on the technology used, and is independent of the profile signaled for either bitstream or audio decoder. In particular, if the profile is changed between one that permits backward compatible decoding and one that requires enhanced decoding, the timestamps and other structures (e.g. edit lists and pre-roll) are not adjusted.

3 AAC Encoder/Decoder Behavior

3.1 Example 1: AAC

3.1.1 Overview

Figure 2 shows the AAC encoder and decoder behavior with respect to the association of encoder input blocks, access units (AU), timestamps and decoder output blocks or composition units (CU). Note that the input signal is only two and a fraction blocks long (as indicated by the oscillating waveform). The encoder essentially extends the waveform at both ends to facilitate encoding of the entire waveform. The ISO Base Media File Format "helper" information "pre-roll" and "edit-list" facilitate exact reconstruction of the encoded waveform segment in the case that the compressed data is stored in an MPEG-4 Format file.

The specifics of encoder behavior are non-normative, except that:

- The encoder must produce normative access units.
- The timestamp associated with those access units must be the time of the first sample of the waveform in the corresponding composition unit.

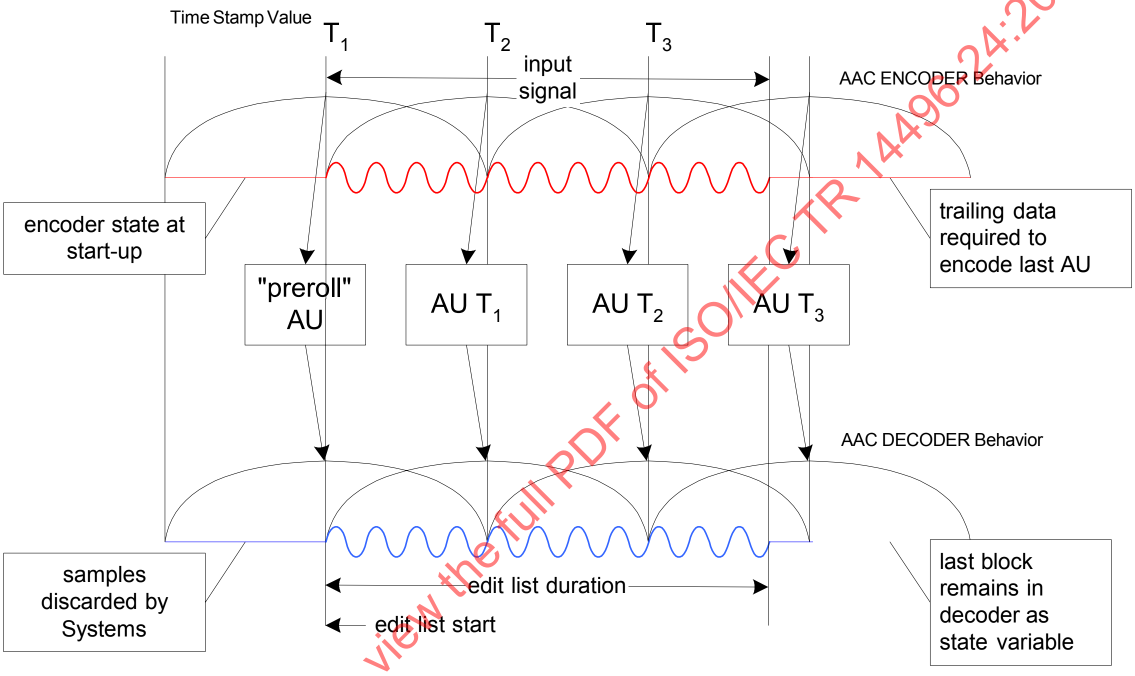

Figure 2 — AAC encoder/decoder behavior

In this example the AAC encoder has a start-up state that represents a virtual 1024 samples that precede the first block of 1024 input samples. This virtual 1024 are concatenated with the first 1024 and then are windowed by the length 2048 window and encoded into one access unit. The window shifts over by 1024, such that the next 1024 samples are shifted in and the oldest 1024 samples are shifted out. This defines the 50% overlap processing that is inherent in the MDCT and is the reason that the figure associates an access unit with a window rather than an input block. Note that some AAC encoders may have a start-up state (or "look-head") that is considerably more than 1024 samples. It is the responsibility of the system that uses the encoder, to transfer to the file the correct information (various encoders add 1024 samples, 2048, or even 2048+64 samples).

On shut-down, the encoder in this example must create an additional one and a fraction blocks of samples (typically filling the remainder of the block with zero data) in order to form the last windowed segment of 2048 for the MDCT. Without creating the trailing portion of the last MDCT window, the encoder would not encode the leading portion of the MDCT window, which is valid data.

The decoder produces a composition unit as output for every access unit it receives as input. The edit-list indicates the desired audio output (that is, the valid samples) from amongst the set of samples in the output composition units. In this example, the edit list specifies that Systems discard the first 1024 audio samples (exactly the result from decoding the pre-roll access unit), and also discard the last 256 samples of the decoded waveform (so that the length of the retained audio segment is 2816 samples). In this way four access units are decoded to obtain an exact representation, within the constraints of lossy coding distortion, of the input waveform. Syntax in the ISO File format can instruct Systems to perform exactly these operation, such that the desired audio, and only the desired audio is obtained.

The ISO File Format syntax can specify the need for “pre-roll”, and in this example the roll-distance value of −1 indicates to Systems that it must start the sequence of access units presented to the decoder with the access unit immediately prior to the access unit whose corresponding compositionBuffer contains the start of the desired audio. This includes the cases of starting at the beginning of the audio (the start of the edit list), random access, or where the user has performed further editing in the encoded domain. The pre-roll syntax is shown in the next section.

3.1.2 Pre-roll

The detailed pre-roll syntax it shown in Annex A. For this example, the pre-roll box would contain:

```
Grouping-type = ‘roll’
Entry-count = 1
Sample-count = <number_of_AUs_in_track>
Group-description-index = 1
Roll-distance = -1
```

This indicates that there is one “pre-roll” group, that one ‘extra’ AU should be supplied to the decoder, and that this applies no matter where the audio starts playback. Note that Sample-count is equal to the number of samples (or access units) in the track.

3.1.3 Edit-list

The detailed edit list syntax it shown in Annex A. For this example, the edit list box would contain:

```
Entry-count = 1
Segment-duration = 35       (movie timescale is typically 1/600 seconds)
Media-Time = 1024    (media timescale is 48000 kHz)
Media-Rate = 1
```

Note that the edit duration is normally expressed in movie timescale units and that the edit start is expressed in media timescale units. The example above indicates that there is one edit, its duration is that of the entire input waveform rounded to the nearest movie timescale value (note that 2816*48000/600 = 35.2), and that the edit begins after the first 1024 samples, or 1024 in media timescale (indicated as “samples discarded by Systems” in Figure 2). Further note that the movie timescale could be changed to be equal to the media timescale (e.g. for audio-only movies), thereby removing the rounding problem when specifying edit duration.

Track duration is an integer that indicates the duration of this track (in the timescale indicated in the Movie Header Box). The value of this field is equal to the sum of the durations of all of the track's edits. If there is no edit list, then the duration is the sum of the sample durations, converted into the timescale in the Movie Header Box. If the duration of this track cannot be determined then duration is set to all 1s (32-bit maxint).

3.1.4 Compressed Information and Decoder behavior

Since encoder behavior is not normative, it may be less confusing to consider the information shown in Figure 3. Here the encoder processing is not indicated, but instead it emphasizes that an access unit has a time stamp, the access unit is decoded into a composition unit, and the timestamp is the time of the first audio sample in that composition unit. Given that normative process, the encoder must behave such that the timestamps on the access units are correct.

The pre-roll and edit-list information carried in the MPEG-4 File then permit the Systems layer to recover the desired decoded audio segment.

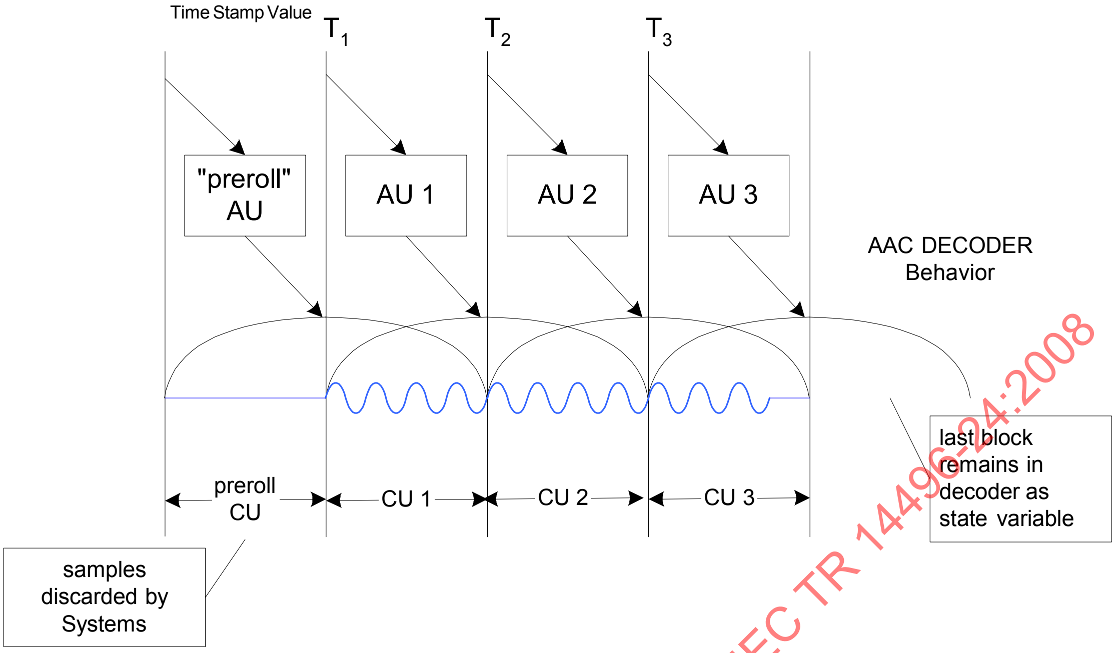

Figure 3 — AAC decoder behaviour

3.2 Example 2: HE-AAC

3.2.1 Overview

Figure 4 shows the HE AAC decoder behavior with respect to the access units and associated composition units. An HE-AAC decoder is essentially an AAC decoder followed by an SBR “post-processing” stage. The additional delay imposed by the SBR tool is due to the QMF bank and the data buffers within the SBR tool. It can be derived by the following.

$$Delay_{SBR-Tool} = L_{AnlysisFilter} - N_{AnalysisChannles} + 1 + Delay_{buffer}$$

where

$$N_{AnalysisChannles} = 32\text{, } L_{AnalysisFilter} = 320 \text{ and } Delay_{buffer} = 6 \cdot 32\text{ .}$$

This means that the delay imposed by the SBR tool (at the input sampling rate, i.e., the output sampling rate of the AAC) is

$$Delay_{SBR-Tool} = 320 - 32 + 1 + 6 \cdot 32 = 481$$

samples. Typically, the SBR tool runs in the “upsampling” (or “dual rate”) mode, in which case the 481 sample delay at the AAC sampling rate translates to a 962 sample delay at the SBR output rate. It could also operate at the same sampling rate as the AAC output (denoted as “downsampled SBR mode”), in which case the additional delay is only 481 samples at the SBR output rate. There is a “backwards compatible” mode in which the SBR tool is neglected and the AAC output is the decoder output. In this case there is no additional delay.

Figure 4 shows the decoder behavior for the most common case in which the SBR tool runs in upsampling mode and the additional delay is 962 output samples. This delay corresponds to approx. 47 % of the length of the upsampled AAC frame (after SBR processing). Note that T1 is the timestamp associated with CU 1 after the delay of 962 samples, that is, the timestamp for the first valid sample of HE AAC output. Further note that if HE AAC is running in “downsampled SBR mode” or “single-rate” mode, the delay would be 481 samples but the timestamp would be identical since in single-rate mode the CU’s are half the number of samples so that the delay is still 47 % of the CU duration.

For all of the available signaling mechanisms (i.e., implicit signaling, backward compatible explicit signaling, or hierarchical explicit signaling) if the decoder is HE-AAC then it must convey to Systems any additional delay incurred by SBR processing, otherwise the lack of an indication from the decoder indicates that the decoder is AAC. Hence, Systems can adjust the timestamps so as to compensate for the additional SBR delay.

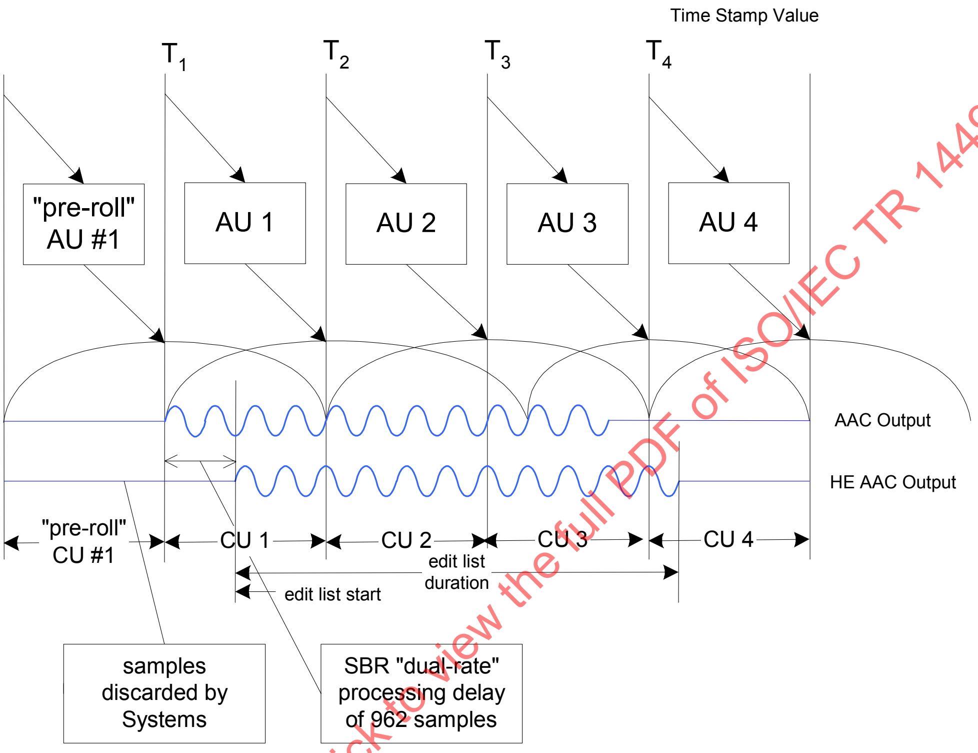

Figure 4 — HE AAC decoder behavior: dual-rate mode

4 Streaming Considerations

Not strictly within the scope of MPEG, it is nonetheless worth discussing how these issues can be handled in RTP streaming. In that context, there is no ‘edit list’ and there is also no ability to warn of the need for pre-roll. The best that can be done is to send all the access units, but with timestamps that respect the edit list.

Consider the following example:

- The time-scale of the audio stream, and the sampling rate, are 48000 Hz.
- The file indicates that the ‘desired’ audio starts at sample 65 in the second access unit.
- The desired time of the start of the audio is R on the RTP time-line.

The system sends the first access unit with a timestamp R-64-1024, and the second with a timestamp of R-64. The client will play out the (undesired) pre-roll data, but the subsequent audio will be in sync with other media, such as video, as is desired.