RTP Payload Format for Phoneme/Facial Animation Parameter (PFAP) Streams

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

2. Abstract

This document describes a Real-Time Transport Protocol (RTP) payload format for transporting phoneme and facial animation parameter (PFAP) streams over the Internet according to the TtsFAPInterface that is defined as an internal interface of an MPEG-4 client in ISO/IEC 14496-3 (MPEG-4 Audio, Subpart 6: Text-to-Speech Interface, TtsFAPInterface) [2]. A recovery strategy for loss-tolerant transmission of such streams is described.

Table of Contents

1. Status of this Memo 1
2. Abstract 1
3. Introduction 3
4. Requirements language 4
5. The MPEG-4 class TtsFAPInterface 4
6. Payload Format 6
6.1. Packet descriptor 7
6.2. Phoneme descriptor 8
6.3. FAP descriptor 9
6.4. Recovery information, type 1 10
7. RTP header fields usage: 11
8. Recovery Strategy 11
9. Security Considerations 11
10. References 13
11. Author's Addresses 13

3. Introduction

Animated talking heads based on MPEG-4 [1] may be implemented on a client that renders the head and synthesizes the speech using a Text-to-Speech (TTS) application on the client. The MPEG-4 standard defines only the input interface and two output interfaces for a compliant TTS application. The output interfaces are supposed to be internal to the MPEG-4 client and, thus, no transport protocol is defined related to transmission of the output data. However, advanced TTS servers may need to be implemented on network-based machines and shared by many users. In order to animate talking heads on a client using a network-based TTS server it will be necessary to stream the outputs of the TTS server to the client.

The input to an MPEG-4 compliant TTS server is the "MPEG-4 audio text-to-speech payload" [2] defined for transmitting text to a TTS server. The TTS server synthesizes speech as an audio signal from the text. The text may contain bookmarks that enable the control of the talking head with facial animation parameters (FAP) synchronized with the speech. FAPs may define facial expressions like joy and disgust, head orientation and other deformations of flexible parts of the head. Bookmarks do not influence the synthesized speech. The "MPEG-4 audio text-to-speech payload" may also transport optional TTS control information like Gender, Age, and Speech_Rate. The "MPEG-4 audio text-to-speech payload" may be transported using the MPEG-4 payload format as specified in [3].

One of the outputs of the TTS server is the audio stream. This audio stream with the related timing information is handed to the compositor of the MPEG-4 client. The compositor enables synchronized playback of MPEG-4 supported media. In a network based TTS server, the compositor will be located at the client side and the audio stream produced by the TTS server needs to be transmitted to the client. Several RTP payload formats for audio streams already exist and may be used in this context.

The other output of the TTS server is the TTS markup information. MPEG-4 defines the class TtsFAPInterface that holds the TTS markup information [2]. This class is used to hand the TTS markup information from the TTS server to the face renderer within the compositor of the MPEG-4 client. The TTS markup information enables an MPEG-4 client to create the animation of the talking head such that the head produces visual speech (mainly lip motion) synchronized with the audio. The TTS markup information contains phonemes, bookmarks, and related timing information.

A phoneme is the basic spoken unit in a language. Pronouncing a phoneme involves coordinating movements of the lungs, vocal cavities, larynx, lips, tongue, and teeth. The TTS server

translates the text to be synthesized into phonemes. Furthermore, the TTS server computes the start time and duration of each phoneme in the synthesized speech.

A bookmark is the exact copy of the bookmark in the text sent to the TTS server. MPEG-4 specifies that the start time of a FAP in a bookmark is the start time of the first phoneme of the first word following the bookmark of the current sentence. If there is no word after the bookmark in the current sentence, the start time of the FAP is the same as the start time of the last phoneme of the previous word. Hence, the start time of a FAP always coincides with a phoneme. MPEG-4 allows up to 40 consecutive bookmarks that can be used to render complicated expressions.

In order to enable networked TTS servers to be used with MPEG-4, a novel payload format for TTS markup information needs to be defined.

In this document we define an RTP payload format for transporting Phoneme/FAP (PFAP) streams over the Internet using RTP. The payload format is based on the TtsFAPInterface defined in Subpart 6 of the ISO/IEC International Standard 14496-3 [2] and outlined in Section 5 of this document. The payload format includes packet loss recovery information.

4. Requirements language

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 [5].

5. The MPEG-4 class TtsFAPInterface

In this section, we describe the class TtsFAPInterface, its parameters and its usage since it is the basic structure carried by the new payload format proposed in this document. The class TtsFAPInterface is used to hand the TTS markup information from the TTS server to the face renderer within the compositor of the MPEG-4 client. This class holds one phoneme and related information, namely PhonemeSymbol, PhonemeDuration, f0Average, Stress, WordBegin, Bookmark, and Starttime.

PhonemeSymbol:
: This field identifies a phoneme using an 8 bit unsigned integer (PhonemeSymbol). A language usually uses around 50 phonemes. Phonemes may be specified by Unicode. Since MPEG-4 uses the class TtsFAPInterface only internally in a client, it does not specify the mapping of a phoneme specified in Unicode to this 8 bit PhonemeSymbol.

PhonemeDuration:
This field identifies the duration of the PhonemeSymbol in units of milliseconds using a 12 bit unsigned integer.

f0Average:
This field defines the frequency of the synthesized audio signal for this phoneme in units of 2 Hz using an 8 bit unsigned integer.

Stress:
Stress indicates a stressed phoneme using 1 bit.

Bookmark:
This field is a string that contains one or more bookmarks that are associated with the current PhonemeSymbol. A definition of the bookmark structure is given in [1], Annex C. A bookmark starts with "<FAP" and ends with ">". Between the start and end strings of a bookmark, there are four fields defined: n (FAP number 2<=n<=68), FAPfield (see below), T (transition time), and C (time curve for computation of the amplitude during the transition time).

In case of n=2, FAPfield holds the four numbers "e1 a1 e2 a2", with the two facial expressions e1 and e2 and their target amplitudes a1 and a2, respectively. There are six different facial expressions (1<= e1,e2<=6) defined in Annex C of [1]. In case of 3<=n<=68, FAPfield holds only the target amplitude "a" for FAP n.

Amplitudes are given in different units. The unit of an amplitude is determined by the FAP n. The maximum value of the amplitude is signed 2529600. It may be reached for head and eye rotations. In these cases, the unit is AU (Angle Units, 0.00001 RAD), and the maximum value corresponds to 25.296 RAD.

There are no limits on the transition time T specified in ms.

The field C can be 1, 2, or 3, which is an identifier for a time curve equation defined in [1], Annex C. The time curve describes the transition of the FAP amplitude from its current amplitude to the target amplitude a (a1 and a2 in case of n=2) of the FAP at the end of the transition time T. The amplitude of the FAP at the beginning of the transition depends on the previous bookmarks and can be equal to:

- 0 if no bookmark with FAP number was used before.
- a of the previous bookmark with the same FAP number if a time longer than the previous transition time T has elapsed between these two FAP bookmarks.
- The actual reached amplitude due to a previous bookmark with the same FAP number if a time shorter

than the previous transition time T has elapsed between the previous bookmark and the current one. At the end of the transition time T, target amplitude a is maintained until another bookmark gives a new target amplitude. To reset a FAP, a bookmark with the same FAP number with a=0 is included in the text.

In case of C=1, the face renderer will linearly change the amplitude of FAP n from its current amplitude to the target amplitude within the transition time T. In case of C=2, a triangle function is used which linearly changes the amplitude of FAP n from its current value to the target amplitude a within the transition time T/2. After that the amplitude is linearly changed back to the value prior to encountering the bookmark within the transition time T/2. In case of C=3, a spline function is used to change the amplitude from its current amplitude to the target amplitude a within the transition time T.

Bookmarks with n=2 allow to change the facial expression of the face (joy, anger, etc.), and n in the range of 3 to 68 allow to animate parts of the head (lips, eyebrow, etc.)

Starttime:
Start time for this phonemeSymbol with respect to the start of the MPEG-4 session in ms using a long int. MPEG-4 computes the duration of the phonemes by subtracting the start times of consecutive phonemes. In the PFAP payload format, we transmit time durations with each phoneme.

6. Payload Format

The PFAP payload consists of three types of information: phoneme descriptor, FAP descriptor, and recovery information. Each payload starts with a “packet descriptor” field followed by optional recovery information. Phoneme descriptors and FAP descriptors may follow the packet descriptor or the recovery information if available. FAPs are associated with phonemes to determine their timing in a sentence (see section 3, or [2]). The start time of a FAP is the same as the start time of the first phoneme following the FAP(s). In case that the input to the TTS server ends with a bookmark, the server could send these bookmarks as FAPs prior to the last phoneme of the previous word. Alternatively, the server could create a short silence phoneme that is sent after the final FAP. Therefore, a packet MUST end with a phoneme if it contains any information other than recovery information.

The following sections define the specific formats for the packet descriptor and each of the three information types.

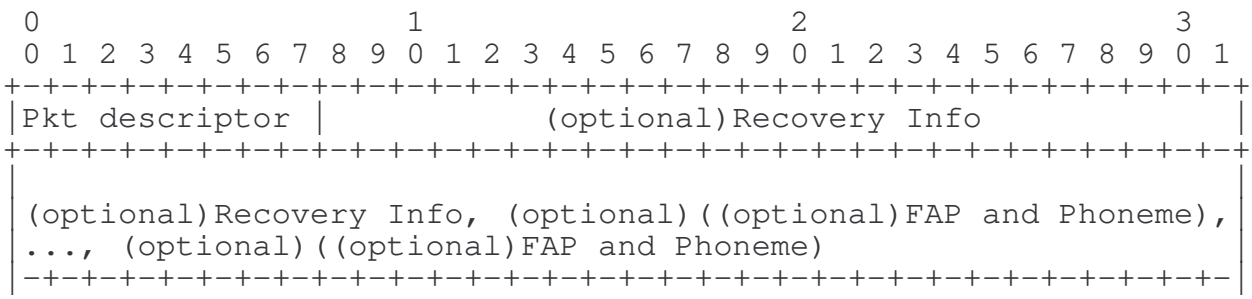

Figure 1 - PFAP Payload

6.1. Packet descriptor

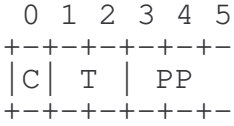

Figure 2 - Packet descriptor

Complete (C): 1 bit
Distinguish between dynamic, and complete recovery information. Zero stands for dynamic, and one for complete recovery information. In case of complete recovery information, the packet MUST only contain recovery information. Recovery information is defined in section “6.4 Recovery information, type 1”.

Type (T): 2 bits
This field identifies the structure of recovery information with the following meaning:
- 00 no recovery information
- 01 recovery information (defined in “6.4 Recovery information, type 1”)
- 10 reserved
- 11 reserved

prevPackets (PP): 3 bits
For dynamical recovery (C=0) this field defines the number of previous packets that can be recovered with the following recovery information. For complete recovery information (C=1) this field can be ignored. The interpretation of these three bits is given as follows:
- 000 reserved
- 001 one previous packet is covered
- 010 two previous packets are covered
- 011 four previous packets are covered
- 100 seven previous packets are covered

101 15 previous packets are covered
110 25 previous packets are covered
111 40 previous packets are covered

InfoBits (IB): 2 bits
Indicate the type of the descriptor following the recovery information:
00 a Phoneme descriptor follows
01 a FAP descriptor follows
10 end of packet
11 reserved

6.2. Phoneme descriptor

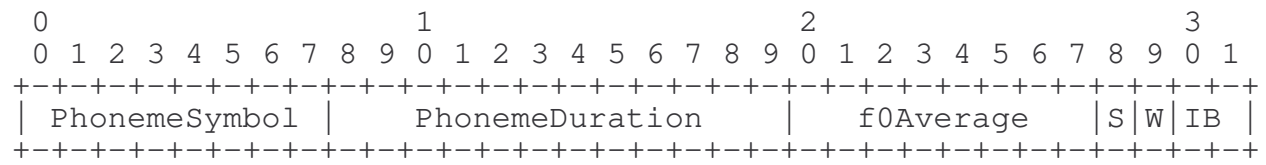

Figure 3 - Phoneme descriptor

PhonemeSymbol: 8 bits
This field identifies each phoneme from a phoneme alphabet. The mapping of Phonemes to this 8 bit number is signaled out of band.

PhonemeDuration: 12 bits
This field identifies the duration of the PhonemeSymbol in units of milliseconds.

f0Average: 8 bits
This field defines the frequency of the synthesized audio signal for this phoneme in units of 2 Hz.

Stress (S): 1 bit
S=1 indicates a stressed phoneme.

WordBegin (W): 1 bit
W=1 indicates the beginning of a word.

InfoBits (IB): 2 bits
These bits identify the following descriptor (phoneme, FAP) in the stream or indicate the end of text, which can be after a sentence or a paragraph. The meanings of the binary combinations are:
00 a Phoneme descriptor follows
01 a FAP descriptor follows
10 end of packet

11 end of text, which implies end of packet. End of text can be at the end of a sentence or paragraph. The renderer/client should expect a pause of undefined length prior to the next utterance.

6.3. FAP descriptor

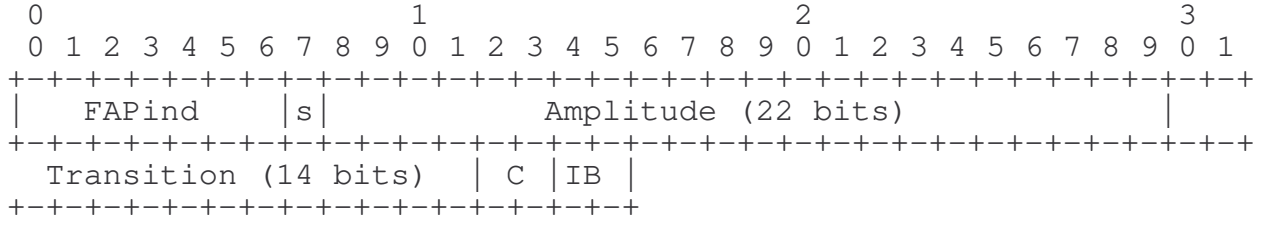

Figure 4 - FAP descriptor

FAPind: 7 bits
This field identifies FAPs in the range of 3 to 74. Facial expressions are indicated using FAP numbers larger than 68. For FAP numbers larger than 68, (FAP number - 68) gives the facial expression number e1 or e2. Amplitude, transition and curve are not mapped, they stay the same.
Example:
bookmark sequence for expression: <FAP 2 1 40 2 30 2000 3>
transformed bookmark sequence: <FAP 69 40 2000 3><FAP 70 30 2000 3>

Sign (s): 1 bit
Sign of the FAP target amplitude. 0 stands for plus, and 1 for minus.

Amplitude: 22 bits
This field holds the target amplitude for this FAP. The maximum possible target amplitude is 2529600.

Transition: 14 bits
Holds the desired transition time during which the target amplitude of the FAP has to be reached. The maximum transition time is not specified in MPEG-4. In this payload format, it is limited to 16383 ms.

Curve (C): 2 bits
Describes the time curve (1, 2, or 3) used for computation of the FAP amplitude.

InfoBits (IB): 2 bits

These bits identify the following descriptor (phoneme, FAP) in the stream. The meanings of the binary combinations are:

00 a Phoneme descriptor follows
01 a FAP descriptor follows
10 reserved
11 reserved

6.4. Recovery information, type 1

Only FAPs can be recovered with the recovery information. In case of complete recovery information, only FAPs with nonzero amplitudes are specified. In case of dynamic recovery, only FAPs from bookmarks that were specified during the prevPackets packets and still have an effect on the FAPs are specified. This might include FAPs with a target amplitude of 0. As an example, if a FAP is changed during a previous packet using a triangle function (C=2) and the transition time is already in the past, the FAP is not included in the recovery bit structure.

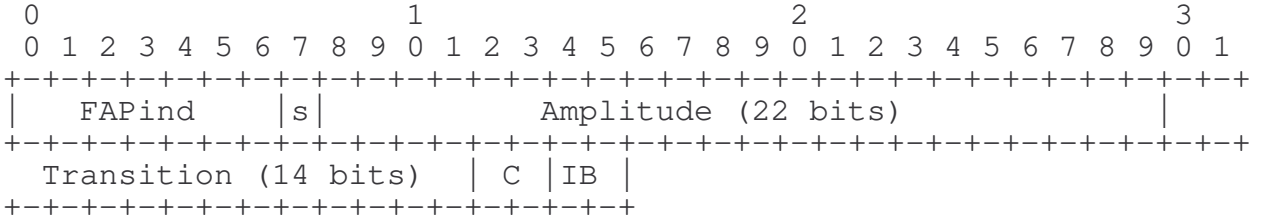

Figure 5 – Recovery information, type 1

FAPind: 7 bits
see FAP descriptor in “6.3 FAP descriptor”

Sign (s): 1 bit
see FAP descriptor in “6.3 FAP descriptor”

Amplitude: 22 bits
see FAP descriptor in “6.3 FAP descriptor”

Transition: 14 bits
Holds the transition time adjusted for the moment of sending of each transmitted FAP. This new transition time should be set to the greater of 0 or the end time of transition minus the timestamp of the packet.

Curve (C): 2 bits
see FAP descriptor in “6.3 FAP descriptor”

InfoBits (IB): 2 bits
These Bits are describing the following data.

The meanings of the binary combinations are:

00	recovery information, type 1 follows
01	reserved
10	reserved
11	indicates the end of recovery information

7. RTP header fields usage:

Payload Type: The assignment of an RTP payload type for this payload format is outside the scope of this document, and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this format, or if that is not done then a payload type in the dynamic range shall be chosen.

M bit: Marker Bit equals one indicates the start of a sentence with the first phoneme in the current packet. This non-speech related information is to be used with the renderer.

Timestamp: Represents the presentation time of the first phoneme in this packet based on a 44.1 kHz clock unless specified otherwise out-of-band. For packets without phonemes (complete recovery) the timestamp specifies the time when the state of the bookmarks was sampled.

8. Recovery Strategy

Recovery information is sent using the 6.4 Recovery information, type 1. Complete recovery information MAY be sent between two regular data packets. Dynamical recovery information MAY be sent with each regular data packet. Dynamical recovery information contains FAPs that were transmitted during the recovery period prevPackets. Complete recovery only contains non-zero FAPs. Complete recovery packets are only sent for new clients/users or burst losses exceeding the limits of dynamical recovery.

9. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [5], and any appropriate profile. This implies that confidentiality of the media streams is achieved by encryption. Because the data encoding used with this payload format is applied end-to-end, encryption may be performed after encoding so there is no conflict between the two operations.

A potential denial-of-service threat exists for data encodings using receiver side decoding. The attacker can inject pathological

datagrams into the stream, which are complex to decode and cause the receiver to be overloaded. The decoder software should consider this possibility and take the necessary precautions.

As with any IP-based protocol, in some circumstances, a receiver may be overloaded simply by the receipt of too many packets, either desired or undesired. Network-layer authentication may be used to discard packets from undesired sources, but the processing cost of the authentication itself may be too high. In a multicast environment, pruning of specific sources may be implemented in future versions of IGMP [6] and in multicast routing protocols to allow a receiver to select which sources are allowed to reach it.

10. References

[1] ISO/IEC International Standard 14496-2; "Generic coding of audio-visual objects - Part 2: Visual", 1998

[2] ISO/IEC International Standard 14496-3; "Generic coding of audio-visual objects - Subpart 6: Text-to-Speech Interface", 1998

[3] Avaro, et. al., "RTP Payload Format for MPEG-4 Streams", IETF work in progress, draft-ietf-avt-mpeg4-multisl-00.txt, June 2001.

[4] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

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

[6] Deering, S., "Host Extensions for IP Multicasting", STD 5, RFC 1112, August 1989.

11. Author's Addresses

Joern Ostermann
AT&T Labs - Research, Rm A5-4E02
200 Laurel Ave South Phone: 1-732-420-9116
Middletown, NJ 07748 USA Email:
osterman@research.att.com

Juergen Th. Rurainsky
AT&T Labs - Research, Rm A5-4F27
200 Laurel Ave South Phone: 1-732-420-9138
Middletown, NJ 07748 USA Email: jru@research.att.com

M. Reha Civanlar
AT&T Labs - Research, Rm A5-4D04
200 Laurel Ave South Phone: 1-732-420-9170
Middletown, NJ 07748 USA Email:
civanlar@research.att.com