Abstract

This memo describes a Real-time Transport Protocol (RTP) payload format for Timed Text Markup Language (TTML), an XML-based timed text format from W3C. This payload format is specifically targeted at streaming workflows using TTML.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8759.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Table of Contents

1. Introduction
2. Conventions and Definitions
3. Media Format Description
   3.1. Relation to Other Text Payload Types
   3.2. TTML2
4. Payload Format
   4.1. RTP Header Usage
   4.2. Payload Data
5. Payload Content Restrictions
6. Payload Processing Requirements
   6.1. TTML Processor Profile
      6.1.1. Feature Extension Designation
      6.1.2. Processor Profile Document
      6.1.3. Processor Profile Signalling
7. Payload Examples
8. Fragmentation of TTML Documents
9. Protection against Loss of Data
10. Congestion Control Considerations
11. Payload Format Parameters
   11.1. Clock Rate
   11.2. Session Description Protocol (SDP) Considerations
      11.2.1. Examples
12. IANA Considerations
13. Security Considerations
14. Normative References
15. Informative References
1. Introduction

TTML (Timed Text Markup Language) [TTML2] is a media type for describing timed text, such as closed captions and subtitles in television workflows or broadcasts, as XML. This document specifies how TTML should be mapped into an RTP stream in streaming workflows, including (but not restricted to) those described in the television-broadcast-oriented European Broadcasting Union Timed Text (EBU-TT) Part 3 [TECH3370] specification. This document does not define a media type for TTML but makes use of the existing application/ttml+xml media type [TTML-MTPR].

2. Conventions and Definitions

Unless otherwise stated, the term "document" refers to the TTML document being transmitted in the payload of the RTP packet(s).

The term "word" refers to a data word aligned to a specified number of bits in a computing sense and not to linguistic words that might appear in the transported text.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Media Format Description

3.1. Relation to Other Text Payload Types

Prior payload types for text are not suited to the carriage of closed captions in television workflows. "RTP Payload for Text Conversation" [RFC4103] is intended for low data rate conversation with its own session management and minimal formatting capabilities. "Definition of Events for Modem, Fax, and Text Telephony Signals" [RFC4734] deals in large parts with the control signalling of facsimile and other systems. "RTP Payload Format for 3rd Generation Partnership Project (3GPP) Timed Text" [RFC4396] describes the carriage of a timed text format with much more restricted formatting capabilities than TTML. The lack of an existing format for TTML or generic XML has necessitated the creation of this payload format.
3.2. TTML2

TTML2 (Timed Text Markup Language, Version 2) [TTML2] is an XML-based markup language for describing textual information with associated timing metadata. One of its primary use cases is the description of subtitles and closed captions. A number of profiles exist that adapt TTML2 for use in specific contexts [TTML-MTPR]. These include both file-based and streaming workflows.

4. Payload Format

In addition to the required RTP headers, the payload contains a section for the TTML document being transmitted (User Data Words) and a field for the length of that data. Each RTP payload contains one or part of one TTML document.

A representation of the payload format for TTML is Figure 1.

![Figure 1: RTP Payload Format for TTML](image)

4.1. RTP Header Usage

RTP packet header fields SHALL be interpreted, as per [RFC3550], with the following specifics:

Marker Bit (M): 1 bit

The marker bit is set to "1" to indicate the last packet of a document. Otherwise, set to "0". Note: The first packet might also be the last.

Timestamp: 32 bits

The RTP Timestamp encodes the epoch of the TTML document in User Data Words. Further detail on its usage may be found in Section 6. The clock frequency used is dependent on the application and is specified in the media type rate parameter, as per Section 11.1. Documents spread across multiple packets MUST use the same timestamp but different consecutive Sequence Numbers. Sequential documents MUST NOT use the same timestamp. Because packets do not represent any constant duration, the timestamp cannot be used to directly infer packet loss.
Reserved: 16 bits
These bits are reserved for future use and **MUST** be set to 0x0 and ignored upon reception.

Length: 16 bits
The length of User Data Words in bytes.

User Data Words: The length of User Data Words **MUST** match the value specified in the Length field.
The User Data Words section contains the text of the whole document being transmitted or a part of the document being transmitted. Documents using character encodings where characters are not represented by a single byte **MUST** be serialised in big-endian order, a.k.a., network byte order. Where a document will not fit within the Path MTU, it may be fragmented across multiple packets. Further detail on fragmentation may be found in Section 8.

### 4.2. Payload Data

TTML documents define a series of changes to text over time. TTML documents carried in User Data Words are encoded in accordance with one or more of the defined TTML profiles specified in the TTML registry [TTML-MTPR]. These profiles specify the document structure used, systems models, timing, and other considerations. TTML profiles may restrict the complexity of the changes, and operational requirements may limit the maximum duration of TTML documents by a deployment configuration. Both of these cases are out of scope of this document.

Documents carried over RTP **MUST** conform to the following profile, in addition to any others used.

### 5. Payload Content Restrictions

This section defines constraints on the content of TTML documents carried over RTP.

Multiple TTML subtitle streams **MUST NOT** be interleaved in a single RTP stream.

The TTML document instance’s root `tt` element in the `http://www.w3.org/ns/ttml` namespace **MUST** include a `timeBase` attribute in the `http://www.w3.org/ns/ttml#parameter` namespace containing the value `media`.

This is equivalent to the TTML2 content profile definition document in Figure 2.
6. Payload Processing Requirements

This section defines constraints on the processing of the TTML documents carried over RTP.

If a TTML document is assessed to be invalid, then it **MUST** be discarded. This includes empty documents, i.e., those of zero length. When processing a valid document, the following requirements apply.

Each TTML document becomes active at its epoch E. E **MUST** be set to the RTP Timestamp in the header of the RTP packet carrying the TTML document. Computed TTML media times are offset relative to E, in accordance with Section I.2 of [TTML2].

When processing a sequence of TTML documents, where each is delivered in the same RTP stream, exactly zero or one document **SHALL** be considered active at each moment in the RTP time line. In the event that a document $D_{n-1}$ with $E_{n-1}$ is active, and document $D_n$ is delivered with $E_n$ where $E_{n-1} < E_n$, processing of $D_{n-1}$ **MUST** be stopped at $E_n$ and processing of $D_n$ **MUST** begin.

When all defined content within a document has ended, then processing of the document **MAY** be stopped. This can be tested by constructing the intermediate synchronic document sequence from the document, as defined by [TTML2]. If the last intermediate synchronic document in the sequence is both active and contains no region elements, then all defined content within the document has ended.
As described above, the RTP Timestamp does not specify the exact timing of the media in this payload format. Additionally, documents may be fragmented across multiple packets. This renders the RTCP jitter calculation unusable.

6.1. TTML Processor Profile

6.1.1. Feature Extension Designation

This specification defines the following TTML feature extension designation:

\[urn:ietf:rfc:8759#rtp-relative-media-time\]

The namespace \(urn:ietf:rfc:8759\) is as defined by [RFC2648].

A TTML content processor supports the \#rtp-relative-media-time feature extension if it processes media times in accordance with the payload processing requirements specified in this document, i.e., that the epoch E is set to the time equivalent to the RTP Timestamp, as detailed above in Section 6.

6.1.2. Processor Profile Document

The required syntax and semantics declared in the minimal TTML2 processor profile in Figure 3 MUST be supported by the receiver, as signified by those feature or extension elements whose value attribute is set to required.
Note that this requirement does not imply that the receiver needs to support either TTML1 or TTML2 profile processing, i.e., the TTML2 #profile-full-version-2 feature or any of its dependent features.

6.1.3. Processor Profile Signalling

The codecs media type parameter MUST specify at least one processor profile. Short codes for TTML profiles are registered at [TTML-MTPR]. The processor profiles specified in codecs MUST be compatible with the processor profile specified in this document. Where multiple options exist in codecs for possible processor profile combinations (i.e., separated by | operator), every permitted option MUST be compatible with the processor profile specified in this document. Where processor profiles (other than the one specified in this document) are advertised in the codecs parameter, the requirements of the processor profile specified in this document MAY be signalled, additionally using the + operator with its registered short code.

A processor profile (X) is compatible with the processor profile specified here (P) if X includes all the features and extensions in P (identified by their character content) and the value attribute of each is, at least, as restrictive as the value attribute of the feature or extension in P that has the same character content. The term "restrictive" here is as defined in Section 6 of [TTML2].
7. Payload Examples

Figure 4 is an example of a valid TTML document that may be carried using the payload format described in this document.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<tt xml:lang="en"
xmlns="http://www.w3.org/ns/ttml"
xmllns:ttm="http://www.w3.org/ns/ttml#metadata"
xmllns:ttp="http://www.w3.org/ns/ttml#parameter"
xmllns:tts="http://www.w3.org/ns/ttml#styling"
ttp:timeBase="media"
>
  <head>
    <metadata>
      <ttm:title>Timed Text TTML Example</ttm:title>
      <ttm:copyright>The Authors (c) 2006</ttm:copyright>
    </metadata>
    <styling>
      <!-- s1 specifies default color, font, and text alignment -->
      <style xml:id="s1"
        tts:color="white"
        tts:fontFamily="proportionalSansSerif"
        tts:fontSize="100%"
        tts:textAlign="center"
      />
    </styling>
    <layout>
      <region xml:id="subtitleArea"
        style="s1"
        tts:extent="78% 11%"
        tts:padding="1% 5%"
        tts:backgroundColor="black"
        tts:displayAlign="after"
      />
    </layout>
  </head>
  <body region="subtitleArea">
    <div>
      <p xml:id="subtitle1" dur="5.0s" style="s1">
        How truly delightful!
      </p>
    </div>
  </body>
</tt>
```

Figure 4: Example TTML Document
8. Fragmentation of TTML Documents

Many of the use cases for TTML are low bit-rate with RTP packets expected to fit within the Path MTU. However, some documents may exceed the Path MTU. In these cases, they may be split between multiple packets. Where fragmentation is used, the following guidelines MUST be followed:

- It is RECOMMENDED that documents be fragmented as seldom as possible, i.e., the least possible number of fragments is created out of a document.

- Text strings MUST split at character boundaries. This enables decoding of partial documents. As a consequence, document fragmentation requires knowledge of the UTF-8/UTF-16 encoding formats to determine character boundaries.

- Document fragments SHOULD be protected against packet losses. More information can be found in Section 9.

When a document spans more than one RTP packet, the entire document is obtained by concatenating User Data Words from each consecutive contributing packet in ascending order of Sequence Number.

As described in Section 6, only zero or one TTML document may be active at any point in time. As such, there MUST only be one document transmitted for a given RTP Timestamp. Furthermore, as stated in Section 4.1, the marker bit MUST be set for a packet containing the last fragment of a document. A packet following one where the marker bit is set contains the first fragment of a new document. The first fragment might also be the last.

9. Protection against Loss of Data

Consideration must be devoted to keeping loss of documents due to packet loss within acceptable limits. What is deemed acceptable limits is dependent on the TTML profile(s) used and use case, among other things. As such, specific limits are outside the scope of this document.

Documents MAY be sent without additional protection if end-to-end network conditions guarantee that document loss will be within acceptable limits under all anticipated load conditions. Where such guarantees cannot be provided, implementations MUST use a mechanism to protect against packet loss. Potential mechanisms include Forward Error Correction (FEC) [RFC5109], retransmission [RFC4588], duplication [ST2022-7], or an equivalent technique.

10. Congestion Control Considerations

Congestion control for RTP SHALL be used in accordance with [RFC3550] and with any applicable RTP profile, e.g., [RFC3551]. "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions" [RFC8083] is an update to "RTP: A Transport Protocol for Real-time Applications" [RFC3550], which defines criteria for when one is required to stop sending RTP packet streams.
Applications implementing this standard MUST comply with [RFC8083], with particular attention paid to Section 4.4 on Media Usability. [RFC8085] provides additional information on the best practices for applying congestion control to UDP streams.

11. Payload Format Parameters

This RTP payload format is identified using the existing application/ttml+xml media type as registered with IANA [IANA] and defined in [TTML-MTPR].

11.1. Clock Rate

The default clock rate for TTML over RTP is 1000 Hz. The clock rate SHOULD be included in any advertisements of the RTP stream where possible. This parameter has not been added to the media type definition as it is not applicable to TTML usage other than within RTP streams. In other contexts, timing is defined within the TTML document.

When choosing a clock rate, implementers should consider what other media their TTML streams may be used in conjunction with (e.g., video or audio). In these situations, it is RECOMMENDED that streams use the same clock source and clock rate as the related media. As TTML streams may be aperiodic, implementers should also consider the frequency range over which they expect packets to be sent and the temporal resolution required.

11.2. Session Description Protocol (SDP) Considerations

The mapping of the application/ttml+xml media type and its parameters [TTML-MTPR] SHALL be done according to Section 3 of [RFC4855].

- The type name "application" goes in SDP "m=" as the media name.
- The media subtype "ttml+xml" goes in SDP "a=rtpmap" as the encoding name.
- The clock rate also goes in "a=rtpmap" as the clock rate.

Additional format-specific parameters, as described in the media type specification, SHALL be included in the SDP file in "a=fmtp" as a semicolon-separated list of "parameter=value" pairs, as described in [RFC4855]. The codecs parameter MUST be included in the a=fmtp line of the SDP file. Specific requirements for the "codecs" parameter are included in Section 6.1.3.

11.2.1. Examples

A sample SDP mapping is presented in Figure 5.
In this example, a dynamic payload type 112 is used. The 90 kHz RTP timestamp rate is specified in the "a=rtpmap" line after the subtype. The codecs parameter defined in the "a=fmtp" line indicates that the TTML data conforms to Internet Media and Captions (IMSC) 1.1 Text profile [TTML-IMSC1.1].

Figure 5: Example SDP Mapping

In this example, a dynamic payload type 112 is used. The 90 kHz RTP timestamp rate is specified in the "a=rtpmap" line after the subtype. The codecs parameter defined in the "a=fmtp" line indicates that the TTML data conforms to Internet Media and Captions (IMSC) 1.1 Text profile [TTML-IMSC1.1].

12. IANA Considerations

This document has no IANA actions.

13. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] and in any applicable RTP profile, such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals (like confidentiality, integrity, and source authenticity) for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this Security Considerations section discusses the security impacting properties of the payload format itself.

To avoid potential buffer overflow attacks, receivers should take care to validate that the User Data Words in the RTP payload are of the appropriate length (using the Length field).

This payload format places no specific restrictions on the size of TTML documents that may be transmitted. As such, malicious implementations could be used to perform denial-of-service (DoS) attacks. [RFC4732] provides more information on DoS attacks and describes some mitigation strategies. Implementers should take into consideration that the size and frequency of documents transmitted using this format may vary over time. As such, sender implementations should avoid producing streams that exhibit DoS-like behaviour, and receivers should avoid false identification of a legitimate stream as malicious.

As with other XML types and as noted in Section 10 of "XML Media Types" [RFC7303], repeated expansion of maliciously constructed XML entities can be used to consume large amounts of memory, which may cause XML processors in constrained environments to fail.
In addition, because of the extensibility features for TTML and of XML in general, it is possible that "application/ttml+xml" may describe content that has security implications beyond those described here. However, TTML does not provide for any sort of active or executable content, and if the processor follows only the normative semantics of the published specification, this content will be outside TTML namespaces and may be ignored. Only in the case where the processor recognizes and processes the additional content or where further processing of that content is dispatched to other processors would security issues potentially arise. And in that case, they would fall outside the domain of this RTP payload format and the application/ttml+xml registration document.

Although not prohibited, there are no expectations that XML signatures or encryption would normally be employed.

Further information related to privacy and security at a document level can be found in Appendix P of [TTML2].

14. Normative References


15. Informative References


Acknowledgements

Thanks to Nigel Megitt, James Gruessing, Robert Wadge, Andrew Bonney, James Weaver, John Fletcher, Frans de Jong, and Willem Vermost for their valuable feedback throughout the development of this document. Thanks to the W3C Timed Text Working Group and EBU Timed Text Working Group for their substantial efforts in developing the timed text format this payload format is intended to carry.

Author's Address

James Sandford
British Broadcasting Corporation
Dock House, MediaCityUK
Salford
United Kingdom
Phone: +44 30304 09549
Email: james.sandford@bbc.co.uk