Technical Specification Group Services and System Aspects **TSGS#23(04)0434**
Meeting #24, Seoul, South Korea
7-10 June 2004

Source: **TSG-SA WG4**

Title: **CR TS 26.234 on Addition of Release 6 functionality (Release 6)**

Document for: **Approval**

Agenda Item: **7.4.3**

The following CR, agreed at the TSG-SA WG4 meeting #31, are presented to TSG SA #24 for approval.

Spec	CR	Rev	Phase	Subject	Cat	Vers	WG	Meeting	S4 doc
26.234	068	1	Rel-6	Addition of Release 6 functionality	B	5.6.0	S4	TSG-SA WG4#31	S4-040307

3GPP TSG-SA4 Meeting #31
Montreal, Canada, 17-21 May 2004

Tdoc ⌘S4-040307

CR-Form-v7

CHANGE REQUEST

⌘	**26.234**	**CR**	**068**	⌘ **rev**	**1**	⌘	Current version: **5.6.0**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ □ ME ☒ Radio Access Network □ Core Network □

Title:	⌘	Addition of Release-6 functionality		
Source:	⌘	TSG SA WG4		
Work item code:	⌘	PSSrel6-Stage3	***Date:*** ⌘	08/06/2004
Category:	⌘	**B**	***Release:*** ⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change: ⌘ Release-6 functionality added.

Summary of change: ⌘ A number of new features and functions have been included in R6, such as:

- Support for media stream selection (alternatives in SDP)
- Support for bitrate adaptation
- Extended support for synthetic audio
- Support for Quality of Experience
- Clarifications and updates to a number of protocols (SDP, RTSP, RTP)
- Update of PSS UAProf vocabulary and RDF schema for Rel-6.

The following functionalities have been moved to their own TSs in R6:

- The 3GPP file format (3GP), moved to TS 26.244
- The Timed text format, moved to TS 26.245
- The 3GPP SMIL Language Profile, moved to TS 26.246

Consequences if not approved: ⌘ Release 6 will NOT contain an updated streaming feature.

Clauses affected: ⌘ All clauses

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification;

The 3GPP transparent end-to-end packet-switched streaming service (PSS) specification consists of six~~three~~ 3GPP TSs:~~;~~ 3GPP TS 22.233 [1], 3GPP TS 26.233 [2], 3GPP TS 26.244 [50], 3GPP TS 26.245 [51], 3GPP TS 26.246 [52] and the present document. ~~The first TS contains the service requirements for the PSS, the second TS provides an overview of the 3GPP PSS and the present document the details of protocol and codecs used by the service.~~

The TS 22.233 contains the service requirements for the PSS. The TS 26.233 provides an overview of the PSS. The TS 26.244 defines the 3GPP file format (3GP) used by the PSS and MMS services. The TS 26.245 defines the Timed text format used by the PSS and MMS services. The TS 26.246 defines the 3GPP SMIL language profile. The present document provides the details of the protocols and codecs used by the PSS.

The TS 26.244, TS 26.245 and TS 26.246 start with Release 6. Earlier releases of the 3GPP file format, the Timed text format and the 3GPP SMIL language profile can be found in TS 26.234.

Introduction

Streaming refers to the ability of an application to play synchronised media streams like audio and video streams in a continuous way while those streams are being transmitted to the client over a data network.

Applications, which can be built on top of streaming services, can be classified into on-demand and live information delivery applications. Examples of the first category are music and news-on-demand applications. Live delivery of radio and television programs are examples of the second category.

The 3GPP PSS provides a framework for Internet Protocol (IP) based streaming applications in 3G networks.

1 Scope

The present document specifies the protocols and codecs for the PSS within the 3GPP system. Protocols for control signalling, capability exchange~~, scene description~~, media transport, rate adaptation and protection ~~media encapsulations~~ are specified. Codecs for speech, natural and synthetic audio, video, still images, bitmap graphics, vector graphics, timed text and text are specified.

The present document is applicable to IP-~~-~~based packet-~~-~~switched networks.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 22.233: "Transparent End-to-End Packet-switched Streaming Service; Stage 1".

[2] 3GPP TS 26.233: "Transparent end-to-end packet switched streaming service (PSS); General description".

[3] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[4] IETF RFC 1738: "Uniform Resource Locators (URL)", Berners-Lee T., Masinter L. and McCahill M., December 1994.

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

[6] IETF RFC 2327: "SDP: Session Description Protocol", Handley M. and Jacobson V., April 1998.

[7] IETF STD 0006: "User Datagram Protocol", Postel J., August 1980.

[8] IETF STD 0007: "Transmission Control Protocol", Postel J., September 1981.

[9] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications", Schulzrinne H. et al., July 2003.

[10] IETF RFC 3551: "RTP Profile for Audio and Video Conferences with Minimal Control", Schulzrinne H. and Casner S., July 2003.

[11] IETF RFC 3267: "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", Sjoberg J. et al., June 2002.

[12] (void)

[13] IETF RFC 3016: "RTP Payload Format for MPEG-4 Audio/Visual Streams", Kikuchi Y. et al., November 2000.

[14] IETF RFC 2429: "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", Bormann C. et al., October 1998.

[15] IETF RFC 2046: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", Freed N. and Borenstein N., November 1996.

[16] IETF RFC 3236: "The 'application/xhtml+xml' Media Type", Baker M. and Stark P., January 2002.

[17] IETF RFC 2616: "Hypertext Transfer Protocol – HTTP/1.1", Fielding R. et al., June 1999.

[18] (void)~~3GPP TS 26.071: "Mandatory Speech CODEC speech processing functions; AMR Speech CODEC; General description".~~

[19] 3GPP TS 26.101: "Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec frame structure".

[20] 3GPP TS 26.171: "AMR Wideband Speech Codec; General Description".

[21] ISO/IEC 14496-3:2001: "Information technology – Coding of audio-visual objects – Part 3: Audio".

[22] ITU-T Recommendation H.263 (1998): "Video coding for low bit rate communication".

[23] ITU-T Recommendation H.263 – Annex X (2001): "Annex X: Profiles and levels definition".

[24] ISO/IEC 14496-2:2001: "Information technology – Coding of audio-visual objects – Part 2: Visual".

[25] ISO/IEC 14496-2:2001/Amd 2:2002: "Streaming video profile".

[26] ITU-T Recommendation T.81 (1992) | ISO/IEC 10918-1:1993: "Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines".

[27] C-Cube Microsystems: "JPEG File Interchange Format", Version 1.02, September 1, 1992.

[28] W3C Recommendation: "XHTML Basic", http://www.w3.org/TR/2000/REC-xhtml-basic-20001219, December 2000.

[29] ISO/IEC 10646-1:2000: "Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane".

[30] The Unicode Consortium: "The Unicode Standard", Version 3.0 Reading, MA, Addison-Wesley Developers Press, 2000, ISBN 0-201-61633-5.

[31] W3C Recommendation: "Synchronized Multimedia Integration Language (SMIL 2.0)", http://www.w3.org/TR/2001/REC-smil20-20010807/, August 2001.

[32] CompuServe Incorporated: "GIF Graphics Interchange Format: A Standard defining a mechanism for the storage and transmission of raster-based graphics information", Columbus, OH, USA, 1987.

[33] CompuServe Incorporated: "Graphics Interchange Format: Version 89a", Columbus, OH, USA, 1990.

[34] (void)

[35] (void)~~3GPP TS 26.140: "Multimedia Messaging Service (MMS); Media formats and codecs".~~

[36] (void)

[37] (void)~~3GPP TS 26.201: " Speech Codec speech processing functions; AMR Wideband Speech Codec; Frame Structure".~~

[38] IETF RFC 2083: "PNG (Portable Networks Graphics) Specification Version 1.0", Boutell T., et al., March 1997.

[39] ~~W3C Working Draft Recommendation: "CC/PP structure and vocabularies", http://www.w3.org/Mobile/CCPP/Group/Drafts/WD-CCPP-struct-vocab-20010620/, June 2001.~~W3C Recommendation: "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/, January 2004.

[40] WAP UAProf Specification, http://www1.wapforum.org/tech/terms.asp?doc=WAP-248-UAProf-20011020-a.pdf, October 2001.

[41] ~~W3C Candidate Recommendation: "Resource Description Framework (RDF) Schema Specification 1.0", http://www.w3.org/TR/2000/CR-rdf-schema-20000327, March 2000.~~W3C Recommendation: "RDF Vocabulary Description Language 1.0: RDF Schema", http://www.w3.org/TR/2004/REC-rdf-schema-20040210/, February 2004.

[42] W3C Recommendation: "Scalable Vector Graphics (SVG) 1.1 Specification", http://www.w3.org/TR/2003/REC-SVG11-20030114/, January 2003.

[43] W3C Recommendation: "Mobile SVG Profiles: SVG Tiny and SVG Basic", http://www.w3.org/TR/2003/REC-SVGMobile-20030114/, January 2003.

[44] Scalable Polyphony MIDI Specification Version 1.0, RP-34, MIDI Manufacturers Association, Los Angeles, CA, February 2002.

[45] Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP Version 1.0, RP-35, MIDI Manufacturers Association, Los Angeles, CA, February 2002.

[46] "Standard MIDI Files 1.0", RP-001, in "The Complete MIDI 1.0 Detailed Specification, Document Version 96.1", The MIDI Manufacturers Association, Los Angeles, CA, USA, February 1996.

[47] WAP Forum Specification: "XHTML Mobile Profile", http://www1.wapforum.org/tech/terms.asp?doc=WAP-277-XHTMLMP-20011029-a.pdf, October 2001.

[48] (void)~~"Unicode Standard Annex #13: Unicode Newline Guidelines", by Mark Davis. An integral part of The Unicode Standard, Version 3.1.~~

[49] IETF RFC 3266: "Support for IPv6 in Session Description Protocol (SDP)", Olson S., Camarillo G. and Roach A. B., June 2002.

[50] 3GPP TS 26.244: "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)".~~ISO/IEC 14496-12:2003 | 15444-12:2003: "Information technology — Coding of audio-visual objects — Part 12: ISO base media file format" | "Information technology — JPEG 2000 image coding system — Part 12: ISO base media file format".~~

[51] 3GPP TS 26.245: "Transparent end-to-end packet switched streaming service (PSS); Timed text format".~~ISO/IEC 14496-14:2003: "Information technology — Coding of audio-visual objects — Part 14: MP4 file format".~~

[52] 3GPP TS 26.246: "Transparent end-to-end packet switched streaming service (PSS); 3GPP SMIL Language Profile".~~IETF RFC 3556: "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) bandwidth", Casner S., July 2003.~~

[53] IETF RFC 2234: "Augmented BNF for Syntax Specifications: ABNF", Crocker D. and Overell P., November 1997.

[54] IETF RFC 3066: "Tags for Identification of Languages", Alvestrand H., January 2001.

[55] IETF RFC 3556: "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", Casner S., July 2003.

[56] 3GPP TS 23.107: "Quality of Service (QoS) concept and architecture".

[57] IETF Internet Draft: "Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)", Ott J. et al, http://www.ietf.org/internet-drafts/draft-ietf-avt-rtcp-feedback-08.txt, January 2004.

[58] IETF RFC 3611: "RTP Control Protocol Extended Reports (RTCP XR)", Friedman T., Caceres R. and Clark A., November 2003.

[59] IETF RFC 1952: "GZIP file format specification version 4.3", Deutsch P., May 1996.

[60] IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax", Berners-Lee T., Fielding R., Irvine U.C. and Masinter L., August 1998.

[61] IETF RFC 2732: "Format for Literal IPv6 Addresses in URL's", Hinden R., Carpenter B. and Masinter L., December 1999.

[62] IETF RFC 3555: "MIME Type Registration of RTP Payload Formats", Casner S. and Hoschka P., July 2003.

[63] 3GPP TS 26.090: "Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions".

[64] 3GPP TS 26.073: "ANSI-C code for the Adaptive Multi Rate (AMR) speech codec".

[65] 3GPP TS 26.104: "ANSI-C code for the floating-point Adaptive Multi Rate (AMR) speech codec".

[66] 3GPP TS 26.190: "Speech Codec speech processing functions; AMR Wideband speech codec; Transcoding functions".

[67] 3GPP TS 26.173: "ANCI-C code for the Adaptive Multi Rate - Wideband (AMR-WB) speech codec".

[68] 3GPP TS 26.204: "ANSI-C code for the Floating-point Adaptive Multi-Rate Wideband (AMR-WB) speech codec".

[69] IETF RFC 3548: "The Base16, Base32, and Base64 Data Encodings", Josefsson S., Ed., July 2003.

[70] Mobile DLS, MMA specification v1.0. RP-41 Los Angeles, CA, USA. 2004.

[71] Mobile XMF Content Format Specification, MMA specification v1.0., RP-42, Los Angeles, CA, USA. 2004.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

continuous media: media with an inherent notion of time. In the present document speech, audio, video and timed text

discrete media: media that itself does not contain an element of time.In the present document all media not defined as continuous media

device capability description: a description of device capabilities and/or user preferences. Contains a number of capability attributes

device capability profile: same as device capability description

kilobits: 1000 bits

kilobytes: 1024 bytes

presentation description: contains information about one or more media streams within a presentation, such as the set of encodings, network addresses and information about the content

PSS client: client for the 3GPP packet switched streaming service based on the IETF RTSP/SDP and/or HTTP standards, with possible additional 3GPP requirements according to the present document

PSS server: server for the 3GPP packet switched streaming service based on the IETF RTSP/SDP and/or HTTP standards, with possible additional 3GPP requirements according to the present document

scene description: description of the spatial layout and temporal behaviour of a presentation. It can also contain hyperlinks

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [3] and the following apply.

3GP	3GPP file format
AAC	Advanced Audio Coding
~~BIFS~~	~~Binary Format for Scenes~~
CC/PP	Composite Capability / Preference Profiles
DCT	Discrete Cosine Transform
DLS	Downloadable Sounds
GIF	Graphics Interchange Format
HTML	Hyper Text Markup Language
ITU-T	International Telecommunications Union – Telecommunications
JFIF	JPEG File Interchange Format
MIDI	Musical Instrument Digital Interface
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
~~MP4~~	~~MPEG-4 file format~~
PNG	Portable Networks Graphics
PSS	Packet-switched Streaming Service
QCIF	Quarter Common Intermediate Format
RDF	Resource Description Framework
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol
SMIL	Synchronised Multimedia Integration Language
SP-MIDI	Scalable Polyphony MIDI
SVG	Scalable Vector Graphics
UAProf	User Agent Profile
UCS-2	Universal Character Set (the two octet form)
UTF-8	Unicode Transformation Format (the 8-bit form)
~~UTF-16~~	~~Unicode Transformation Format (the 16-bit form)~~
W3C	WWW Consortium
WML	Wireless Markup Language
XHTML	eXtensible Hyper Text Markup Language
XMF	eXtensible Music Format
XML	eXtensible Markup Language

4 System description

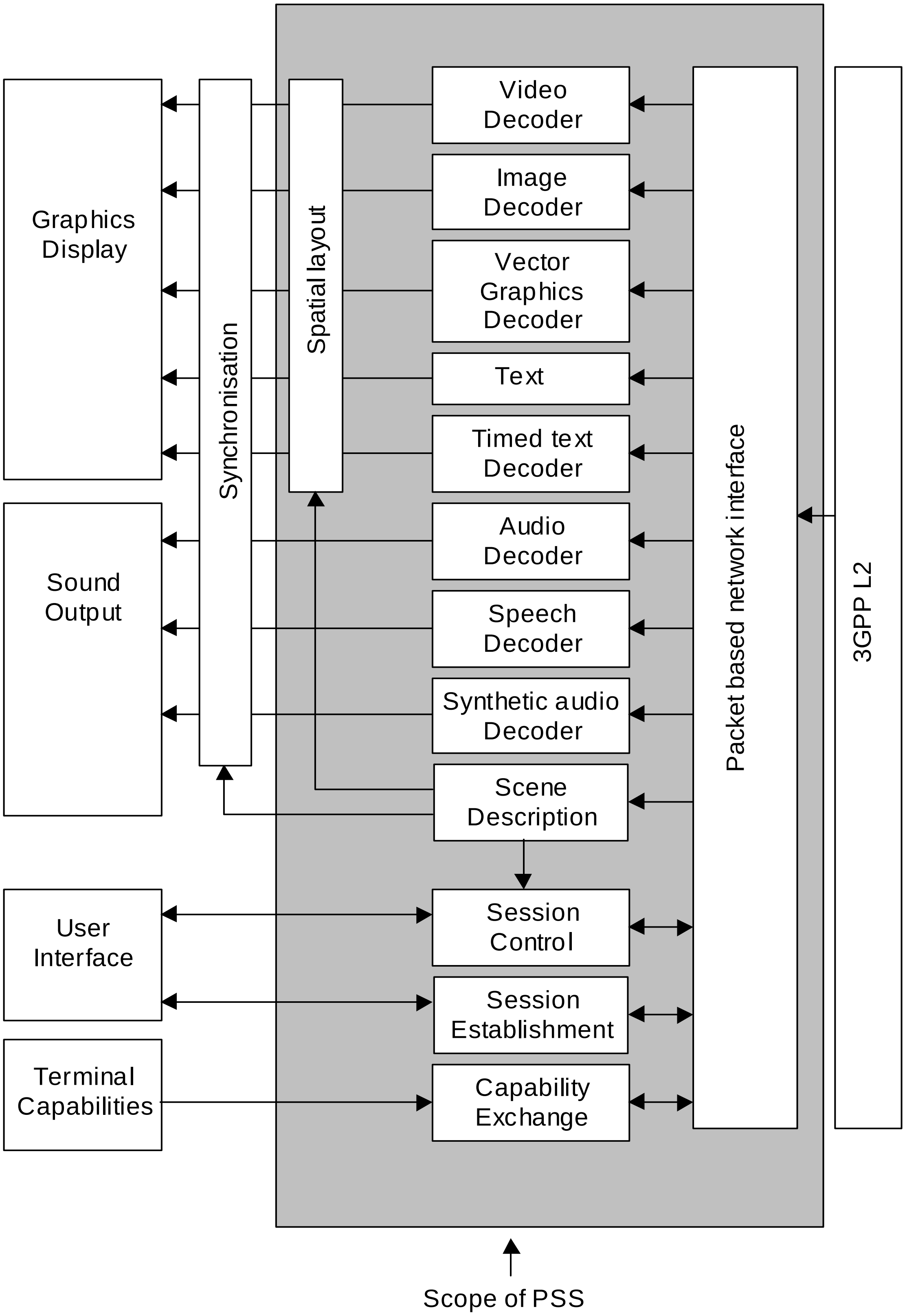


Figure 1: Functional components of a PSS client

Figure 1 shows the functional components of a PSS client. Figure 2 gives an overview of the protocol stack used in a PSS client and also shows a more detailed view of the packet based network interface. The functional components can be divided into control, scene description, media codecs and the transport of media and control data.

The control related elements are session establishment, capability exchange and session control (see clause 5).

- Session establishment refers to methods to invoke a PSS session from a browser or directly by entering an URL in the terminal's user interface.
- Capability exchange enables choice or adaptation of media streams depending on different terminal capabilities.
- Session control deals with the set-up of the individual media streams between a PSS client and one or several PSS servers. It also enables control of the individual media streams by the user. It may involve VCR-like presentation control functions like start, pause, fast forward and stop of a media presentation.

The scene description consists of spatial layout and a description of the temporal relation between different media that is included in the media presentation. The first gives the layout of different media components on the screen and the latter controls the synchronisation of the different media (see clause 8).

The PSS includes media codecs for video, still images, vector graphics, bitmap graphics, text, timed text, natural and synthetic audio, and speech (see clause 7).

Transport of media and control data consists of the encapsulation of the coded media and control data in a transport protocol (see clause 6). This is shown in figure 1 as the "packet based network interface" and displayed in more detail in the protocol stack of figure 2.

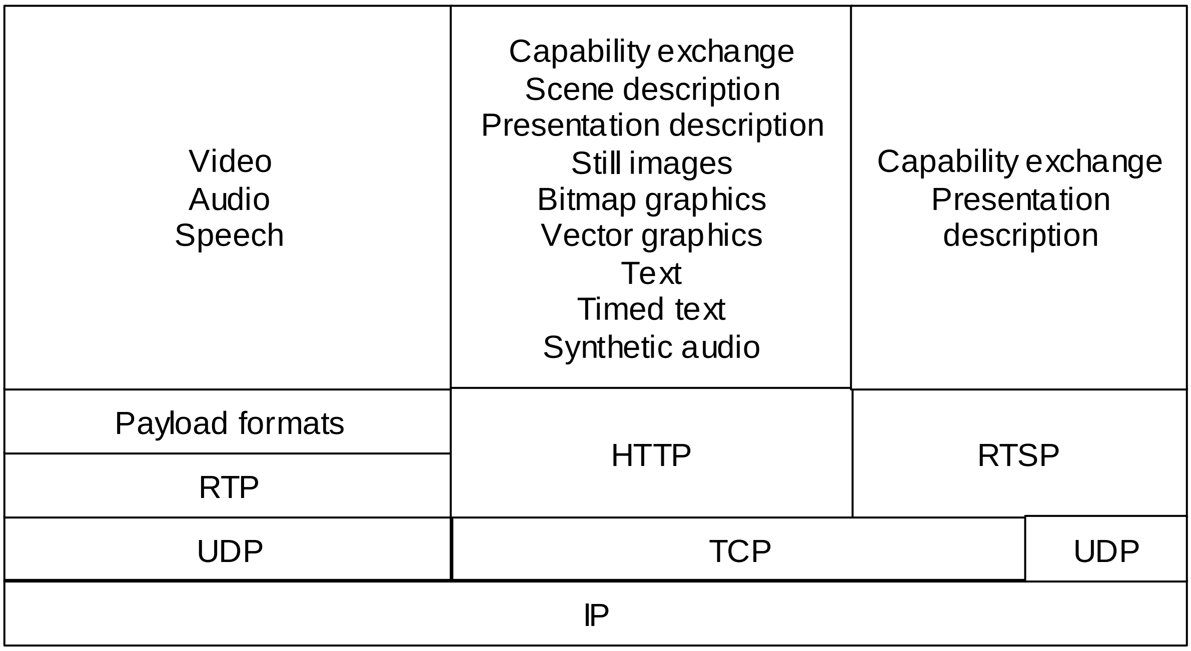


Figure 2: Overview of the protocol stack

5 Protocols

5.1 Session establishment

Session establishment refers to the method by which a PSS client obtains the initial session description. The initial session description can e.g. be a presentation description, a scene description or just an URL to the content.

A PSS client shall support initial session descriptions specified in one of the following formats: SMIL, SDP, or plain RTSP URL.

In addition to rtsp:// the PSS client shall support URLs [4] to valid initial session descriptions starting with file:// (for locally stored files) and http:// (for presentation descriptions or scene descriptions delivered via HTTP).

Examples for valid inputs to a PSS client are: file://temp/morning_news.smil, http://mediaportal/morning_news.sdp, and rtsp://mediaportal/morning_news.

URLs can be made available to a PSS client in many different ways. It is out of the scope of this ~~recommendation~~ specification to mandate any specific mechanism. However, an application using the 3GPP PSS shall at least support URLs of the above type, specified or selected by the user.

The preferred way would be to embed URLs to initial session descriptions within HTML or WML pages. Browser applications that support the HTTP protocol could then download the initial session description and pass the content to the PSS client for further processing. How exactly this is done is an implementation specific issue and out of the scope of this ~~recommendation~~specification.

As an alternative to conventional streaming, a PSS client should also support progressive download of 3GP files [50] delivered via HTTP. A progressive-download session is established with one or more HTTP GET requests. In order to improve playback performance for 3GP files that are not authored for progressive download, a PSS client may issue (multiple pipelined) HTTP GET requests with byte ranges [17]. Example of a valid URL is http://mediaportal/morning_news.3gp.

5.2 Capability exchange

5.2.1 General

Capability exchange is an important functionality in the PSS. It enables PSS servers to provide a wide range of devices with content suitable for the particular device in question. Another very important task is to provide a smooth transition between different releases of PSS. Therefore, PSS clients and servers should support capability exchange.

The specification of capability exchange for PSS is divided into two parts. The normative part contained in clause 5.2 and an informative part in clause A.4 in Annex A of the present document. The normative part gives all the necessary requirements that a client or server shall conform to when implementing capability exchange in the PSS. The informative part provides additional important information for understanding the concept and usage of the functionality. It is recommended to read clause A.4 in Annex A before continuing with clauses 5.2.2-5.2.7.

5.2.2 The device capability profile structure

A device capability profile is an RDF [41] document that follows the structure of the CC/PP framework [39] and the CC/PP application UAProf [40]. Attributes are used to specify device capabilities and preferences. A set of attribute names, permissible values and semantics constitute a CC/PP vocabulary, which is defined by an RDF schema. For PSS, the UAProf vocabulary is reused and an additional PSS specific vocabulary is defined. The details can be found in clause 5.2.3. The syntax of the attributes is defined in the vocabulary schema, but also, to some extent, the semantics. A PSS device capability profile is an instance of the schema (UAProf and/or the PSS specific schema) and shall follow the rules governing the formation of a profile given in the CC/PP specification [39]. The profile schema shall also be governed by the rules defined in UAProf [40] chapter 7, 7.1, 7.3 and 7.4.

5.2.3 Vocabularies for PSS

5.2.3.1 General

Clause 5.2.3 specifies the attribute vocabularies to be used by the PSS capability exchange.

PSS servers should understand the attributes in ~~both~~ the ~~streaming~~ four PSS components of the PSS base vocabulary ~~and~~ as well as the recommended attributes from the UAProf vocabulary [40]. A server may additionally support other UAProf attributes.

5.2.3.2 PSS base vocabulary

The PSS base vocabulary contains ~~one~~ four components called "PssCommon", "Streaming", "3gpFileFormat" and "PssSmil". The division of the vocabulary into these components is motivated by the fact that the PSS contains three different base applications:

- pure RTSP/RTP-based streaming (described by the Streaming component);
- 3GP file download or progressive download (described by the 3gpFileFormat component);
- SMIL presentation (described by the PssSmil component).

The last application can consist of downloadable images, text, etc., as well as RTSP/RTP streaming and downloadable 3GP files. Capabilities that are common to all PSS applications are described by the PssCommon component. The three base applications are distinguished from each other by the source of synchronization: for pure streaming it is RTP, for 3GP files it is inherit in the 3GP file format, and for SMIL presentations timing is provided by the SMIL file.

A vocabulary extension to UAProf shall be defined as an RDF schema. The~~is~~ schema for the PSS base vocabulary can be found in Annex F. T~~The schema t~~ogether with the description of the attributes in the present clause, it defines the vocabulary. The vocabulary is associated with an XML namespace, which combines a base URI with a local XML element name to yield an URI. Annex F provides the details.

~~All PSS attributes are put in a PSS specific component called “Streaming”. The list of PSS attributes is as follows:~~The PSS specific components contain a number of attributes expressing capabilities. The following subclauses list all attributes for each component.

5.2.3.2.1 PssCommon component

Attribute name: **AudioChannels**

Attribute definition: This attribute describes the stereophonic capability of the natural audio device.

Component: ~~Streaming~~PssCommon

Type: Literal

Legal values: “Mono”, “Stereo”

Resolution rule: Locked

EXAMPLE 1: `<AudioChannels>Mono</AudioChannels>`

Attribute name: **MaxPolyphony**

Attribute definition: The MaxPolyphony attribute refers to the maximal polyphony that the synthetic audio device supports as defined in [44].

NOTE: The MaxPolyphony attribute can be used to signal the maximum polyphony capabilities supported by the PSS client. This is a complementary mechanism for the delivery of compatible SP-MIDI content and thus the PSS client is required to support Scalable Polyphony MIDI i.e. Channel Masking defined in [44].

Component: PssCommon~~Streaming~~

Type: Number

Legal values: Integer between 5 and 24

Resolution rule: Locked

EXAMPLE 2: `<MaxPolyphony>8</MaxPolyphony>`

Attribute name: **NumOfGM1Voices**

Attribute definition: The NumOfGM1Voices attribute refers to the maximum number of simultaneous GM1 voices that the synthetic audio engine supports.

Component: PssCommon

Type: Number

Legal values: Integer greater or equal than 5

Resolution rule: Locked

EXAMPLE 3: `<NumOfGMIVoices>24</NumOfGMIVoices>`

Attribute name: **NumOfMobileDLSVoicesWithoutOptionalBlocks**

Attribute definition: The NumOfMobileDLSVoicesWithoutOptionalBlocks attribute refers to the maximum number of simultaneous Mobile DLS [70] voices without optional group of processing blocks that the synthetic audio engine supports.

Component: PssCommon

Type: Number

Legal values: Integer greater or equal than 5

Resolution rule: Locked

EXAMPLE 4: `<NumOfMobileDLSVoicesWithoutOptionalBlocks>24 </NumOfMobileDLSVoicesWithoutOptionalBlocks>`

Attribute name: **NumOfMobileDLSVoicesWithOptionalBlocks**

Attribute definition: The NumOfMobileDLSVoicesWithOptionalBlocks attribute refers to the maximum number of simultaneous Mobile DLS voices with optional group of processing blocks that the synthetic audio engine supports. This attribute is set to zero for devices that do not support the optional group of processing blocks.

Component: PssCommon

Type: Number

Legal values: Integer greater than or equal to 0

Resolution rule: Locked

EXAMPLE 5: `<NumOfMobileDLSVoicesWithOptionalBlocks>24 </NumOfMobileDLSVoicesWithOptionalBlocks>`

Attribute name: **PssVersion**

Attribute definition: Latest PSS version supported by the client.

Component: PssCommon

Type: Literal

Legal values: "3GPP-R4", "3GPP-R5", "3GPP-R6" and so forth.

Resolution rule: Locked

EXAMPLE 6: `<PssVersion>3GPP-R6</PssVersion>`

Attribute name: **RenderingScreenSize**

Attribute definition: The rendering size of the device's screen in unit of pixels available for PSS media presentation. The horizontal size is given followed by the vertical size.

Component: PssCommon

Type: Dimension

Legal values: Two integer values equal or greater than zero. A value equal "0x0"means that there exists no possibility to render visual PSS presentations.

Resolution rule: Locked

EXAMPLE 7: `<RenderingScreenSize>70x15</RenderingScreenSize>`

5.2.3.2.2 Streaming component

Attribute name: **~~Pss~~StreamingAccept**

Attribute definition: List of content types (MIME types) ~~the PSS application supports. Both CcppAccept (SoftwarePlatform, UAProf) and PssAccept can be used but if PssAccept is defined it has precedence over CcppAccept.~~relevant for streaming over RTP supported by the PSS application. Content types listed shall be possible to stream over RTP. For each content type a set of MIME parameters can be specified to signal receiver capabilities. A content type that supports multiple parameter sets may occur several times in the list.

Component: Streaming

Type: Literal (Bag)

Legal values: List of MIME types with related parameters.

Resolution rule: Append

EXAMPLE 1~~3~~:

```
<~~Pss~~StreamingAccept>
  <rdf:Bag>
    <rdf:li>audio/AMR-WB; octet-alignment=1</rdf:li>
    <rdf:li>video/H263-2000; profile=0; level=10~~application/smil~~</rdf:li>
  </rdf:Bag>
</~~Pss~~StreamingAccept>
```

Attribute name: **~~Pss~~StreamingAccept-Subset**

Attribute definition: List of content types for which the PSS application supports a subset. MIME types can in most cases effectively be used to express variations in support for different media types. Many MIME types, e.g. AMR-WB has several parameters that can be used for this purpose. There may exist content types for which the PSS application only supports a subset and this subset cannot be expressed with MIME-type parameters. In these cases the attribute StreamingAccept-Subset is used to describe support for a subset of a specific content type. If a subset of a specific content type is declared in StreamingAccept-Subset, this means that StreamingAccept-Subset has precedence over StreamingAccept. StreamingAccept shall always include the corresponding content types for which StreamingAccept-Subset specifies subsets of.

Subset identifiers and corresponding semantics shall only be defined by the TSG responsible for the present document.~~List of content types for which the PSS application supports a subset. MIME types can in most cases effectively be used to express variations in support for different media types. Many MIME types, e.g. AMR-NB has several parameters that can be used for this purpose. There may exist content types for which the PSS application only supports a subset and this subset can not be expressed with MIME-type parameters. In these cases the attribute PssAccept-Subset is used to describe support for a subset of a specific content type. If~~

~~a subset of a specific content type is declared in PssAccept-Subset, this means that PssAccept-Subset has precedence over both PssAccept and CcppAccept. PssAccept and/or CcppAccept shall always include the corresponding content types for which PSSAccept-Subset specifies subsets of. This is to ensure compatibility with those content servers that do not understand the PssAccept-Subset attribute but do understand e.g. CcppAccept.~~

~~This is illustrated with an example. If PssAccept="audio/AMR", "image/jpeg" and PssAccept-Subset="JPEG-PSS" then "audio/AMR" and JPEG Base line are supported. "image/jpeg" in PssAccept is of no importance since it is related to "JPEG-PSS" in PssAccept-Subset. Subset identifiers and corresponding semantics shall only be defined by the TSG responsible for the present document. The following values are defined:~~

- ~~"JPEG-PSS": Only the two JPEG modes described in clause 7.5 of the present document are supported.~~
- ~~"SVG-Tiny"~~
- ~~"SVG-Basic"~~

Component: Streaming

Type: Literal (Bag)

Legal values: ~~"JPEG-PSS", "SVG-Tiny", "SVG-Basic"~~No subsets defined.

Resolution rule: Append

~~EXAMPLE 4: `<PssAccept-Subset>`~~
~~`  <rdf:Bag>`~~
~~`    <rdf:li>JPEG-PSS</rdf:li>`~~
~~`  </rdf:Bag>`~~
~~`</PssAccept-Subset>`~~

Attribute name: **3gppLinkChar**

Attribute definition: Indicates whether the device supports the 3GPP-Link-Char header according to clause 10.2.1.1.

Component: Streaming

Type: Literal

Legal values: "Yes", "No"

Resolution rule: Override

EXAMPLE 2: `<3gppLinkChar>Yes</3gppLinkChar>`

Attribute name: **AdaptationSupport**

Attribute definition: Indicates whether the device supports client buffer feedback signaling according to clause 10.2.3.

Component: Streaming

Type: Literal

Legal values: "Yes", "No"

Resolution rule: Locked

EXAMPLE 3: `<AdaptationSupport>Yes</AdaptationSupport>`

Attribute name: **ExtendedRtcpReports**

Attribute definition: Indicates whether the device supports extended RTCP reports according to clause 6.2.3.1.

Component: Streaming

Type: Literal

Legal values: "Yes", "No"

Resolution rule: Locked

EXAMPLE 4: `<ExtendedRtcpReports>Yes</ExtendedRtcpReports>`

Attribute name: **MediaAlternatives**

Attribute definition: Indicates whether the device interprets the SDP attributes "alt", "alt-default-id", and "alt-group", defined in clauses 5.3.3.3 and 5.3.3.4.

Component: Streaming

Type: Literal

Legal values: "Yes", "No"

Resolution rule: Override

EXAMPLE 5: `<MediaAlternatives>Yes</MediaAlternatives>`

Attribute name: **RtpProfiles**

Attribute definition: List of supported RTP profiles.

Component: Streaming

Type: Literal (Bag)

Legal values: Profile names registered through the Internet Assigned Numbers Authority (IANA), www.iana.org.

Resolution rule: Append

EXAMPLE 6:

```
<RtpProfiles>
  <rdf:Bag>
    <rdf:li>RTP/AVP</rdf:li>
    <rdf:li>RTP/AVPF</rdf:li>
  </rdf:Bag>
</RtpProfile>
```

~~Attribute name: **PssVersion**~~

~~Attribute definition: PSS version supported by the client.~~

~~Component: Streaming~~

~~Type: Literal~~

~~Legal values: "3GPP-R4", "3GPP-R5" and so forth.~~

~~Resolution rule: Locked~~

~~EXAMPLE 5: `<PssVersion>3GPP-R4</PssVersion>`~~

~~Attribute name: **RenderingScreenSize**~~

~~Attribute definition: The rendering size of the device's screen in unit of pixels. The horizontal size is given followed by the vertical size.~~

~~Component: Streaming~~

~~Type: Dimension~~

~~Legal values: Two integer values equal or greater than zero. A value equal "0x0"means that there exists no possibility to render visual PSS presentations.~~

~~Resolution rule: Locked~~

~~EXAMPLE 6: `<RenderingScreenSize>70x15</RenderingScreenSize>`~~

~~Attribute name: **SmilBaseSet**~~

~~Attribute definition: Indicates a base set of SMIL 2.0 modules that the client supports.~~

~~Component: Streaming~~

~~Type: Literal~~

~~Legal values: Pre-defined identifiers. "SMIL-3GPP-R4" indicates all SMIL 2.0 modules required for scene description support according to clause 8 of Release 4 of TS 26.234. "SMIL-3GPP-R5" indicates all SMIL 2.0 modules required for scene description support according to clause 8 of the present document (Release 5 of TS 26.234).~~

~~Resolution rule: Locked~~

~~EXAMPLE 7: `<SmilBaseSet>SMIL-3GPP-R4</SmilBaseSet>`~~

~~Attribute name: **SmilModules**~~

~~Attribute definition: This attribute defines a list of SMIL 2.0 modules supported by the client. If the SmilBaseSet is used those modules do not need to be explicitly listed here. In that case only additional module support needs to be listed.~~

~~Component: Streaming~~

~~Type: Literal (Bag)~~

~~Legal values: SMIL 2.0 module names defined in the SMIL 2.0 recommendation [31], section 2.3.3, table 2.~~

~~Resolution rule: Append~~

~~EXAMPLE 8:~~

```
<SmilModules>
  <rdf:Bag>
    <rdf:li>BasicTransitions</rdf:li>
    <rdf:li>MulitArcTiming</rdf:li>
  </rdf:Bag>
</SmilModules>
```

Attribute name: **VideoDecodingByteRate**

Attribute definition: If Annex G is not supported, the attribute has no meaning. If Annex G is supported, this attribute defines the peak decoding byte rate the PSS client is able to support. In other words, the PSS client fulfils the requirements given in Annex G with the signalled peak decoding byte rate. The values are given in bytes per second and shall be greater than or equal to 8000. According to Annex G, 8000 is the default peak decoding byte rate for the mandatory video codec profile and level (H.263 Profile 0 Level 10).

Component: Streaming

Type: Number

Legal values: Integer value greater than or equal to 8000.

Resolution rule: Locked

EXAMPLE 7~~9~~: <VideoDecodingByteRate>16000</VideoDecodingByteRate>

Attribute name: **VideoInitialPostDecoderBufferingPeriod**

Attribute definition: If Annex G is not supported, the attribute has no meaning. If Annex G is supported, this attribute defines the maximum initial post-decoder buffering period of video. Values are interpreted as clock ticks of a 90-kHz clock. In other words, the value is incremented by one for each 1/90 000 seconds. For example, the value 9000 corresponds to 1/10 of a second initial post-decoder buffering.

Component: Streaming

Type: Number

Legal values: Integer value equal to or greater than zero.

Resolution rule: Locked

EXAMPLE 8~~10~~: <VideoInitialPostDecoderBufferingPeriod>9000
</VideoInitialPostDecoderBufferingPeriod>

Attribute name: **VideoPreDecoderBufferSize**

Attribute definition: This attribute signals if the optional video buffering requirements defined in Annex G are supported. It also defines the size of the hypothetical pre-decoder buffer defined in Annex G. A value equal to zero means that Annex G is not supported. A value equal to one means that Annex G is supported. In this case the size of the buffer is the default size defined in Annex G. A value equal to or greater than the default buffer size defined in Annex G means that Annex G is supported and sets the buffer size to the given number of octets.

Component: Streaming

Type: Number

Legal values: Integer value equal to or greater than zero. Values greater than one but less than the default buffer size defined in Annex G are not allowed.

Resolution rule: Locked

EXAMPLE 9~~11~~: <VideoPreDecoderBufferSize>30720</VideoPreDecoderBufferSize>

5.2.3.2.3 3gpFileFormat component

Attribute name: **Brands**

Attribute definition: List of supported 3GP profiles identified by brand.

Component: 3gpFileFormat

Type: Literal (Bag)

Legal values: Brand identifiers according to 5.3.4 and 5.4 in [50].

Resolution rule: Append

EXAMPLE 1:

```
<Brands>
  <rdf:Bag>
    <rdf:li>3gp4</rdf:li>
    <rdf:li>3gp5</rdf:li>
    <rdf:li>3gp6</rdf:li>
    <rdf:li>3gr6</rdf:li>
  </rdf:Bag>
</Brands>
```

Attribute name: **3gpAccept**

Attribute definition: List of content types (MIME types) that can be included in a 3GP file and handled by the PSS application. For each content type a set of supported parameters can be given. A content type that supports multiple parameter sets may occur several times in the list. A 3GP file may include timed text [51] and to declare support for this format an identifier ("Timed-Text") shall be used, since no MIME type exists.

Component: 3gpFileFormat

Type: Literal (Bag)

Legal values: List of MIME types with related parameters and the "Timed-Text" identifier.

Resolution rule: Append

EXAMPLE 2:

```
<3gpAccept>
  <rdf:Bag>
    <rdf:li>video/H263-2000; profile=0; level=10</rdf:li>
    <rdf:li>audio/AMR</rdf:li>
    <rdf:li>Timed-Text</rdf:li>
  </rdf:Bag>
</3gpAccept>
```

Attribute name: **3gpAccept-Subset**

Attribute definition: List of content types for which the PSS application supports a subset. MIME types can in most cases effectively be used to express variations in support for different media types. Many MIME types have several parameters that can be used for this purpose. There may exist content types for which the PSS application only supports a subset and this subset cannot be expressed with MIME-type parameters. In these cases the attribute 3gpAccept-Subset is used to describe support for a subset of a specific content type. If a subset of a specific content type is declared in 3gpAccept-Subset, this means that 3gpAccept-Subset has precedence over 3gpAccept. 3gpAccept shall always include the corresponding content types for which 3gpAccept-Subset specifies subsets of.

Subset identifiers and corresponding semantics shall only be defined by the TSG responsible for the present document.

Component: 3gpFileFormat

Type: Literal (Bag)

Legal values: No subsets defined.

Resolution rule: Append

5.2.3.2.4 PssSmil component

Attribute name: **SmilAccept**

Attribute definition: List of content types (MIME types) that can be part of a SMIL presentation. The content types included in this attribute can be rendered in a SMIL presentation. If video/3gpp (or audio/3gpp) is included, downloaded 3GP files can be included in a SMIL presentation. Details on the 3GP file support can then be found in the 3gpFileFormat component. If the identifier "Streaming-Media" is included, streaming media can be included in the SMIL presentation. Details on the streaming support can then be found in the Streaming component. For each content type a set of supported parameters can be given. A content type that supports multiple parameter sets may occur several times in the list.

Component: PssSmil

Type: Literal (Bag)

Legal values: List of MIME types with related parameters and the "Streaming-Media" identifier.

Resolution rule: Append

EXAMPLE 1:

```
<SmilAccept>
  <rdf:Bag>
    <rdf:li>image/gif</rdf:li>
    <rdf:li>image/jpeg</rdf:li>
    <rdf:li>Streaming-Media</rdf:li>
  </rdf:Bag>
</SmilAccept>
```

Attribute name: **SmilAccept-Subset**

Attribute definition: List of content types for which the PSS application supports a subset. MIME types can in most cases effectively be used to express variations in support for different media types. Many MIME types have several parameters that can be used for this purpose. There may exist content types for which the PSS application only supports a subset and this subset cannot be expressed with MIME-type parameters. In these cases the attribute SmilAccept-Subset is used to describe support for a subset of a specific content type. If a subset of a specific content type is declared in SmilAccept-Subset, this means that SmilAccept-Subset has precedence over SmilAccept. SmilAccept shall always include the corresponding content types for which SmilAccept-Subset specifies subsets of.

The following values are defined:

- "JPEG-PSS": Only the two JPEG modes described in clause 7.5 of the present document are supported.
- "SVG-Tiny"
- "SVG-Basic"

Subset identifiers and corresponding semantics shall only be defined by the TSG responsible for the present document.

Component: PssSmil

Type: Literal (Bag)

Legal values: "JPEG-PSS", "SVG-Tiny", "SVG-Basic"

Resolution rule: Append

EXAMPLE 2:

```
<SmilAccept-Subset>
  <rdf:Bag>
    <rdf:li>JPEG-PSS</rdf:li>
    <rdf:li>SVG-Tiny</rdf:li>
  </rdf:Bag>
</SmilAccept-Subset>
```

Attribute name: **SmilBaseSet**

Attribute definition: Indicates a base set of SMIL 2.0 modules that the client supports.

Component: Streaming

Type: Literal

Legal values: Pre-defined identifiers. "SMIL-3GPP-R4" and "SMIL-3GPP-R5" indicate all SMIL 2.0 modules required for scene description support according to clause 8 of Release 4 and Release 5, respectively, of TS 26.234. "SMIL-3GPP-R6" indicates all SMIL 2.0 modules required for scene-description support according to clause 8 of the present document (Release 6 of TS 26.234) and to Release 6 of TS 26.246 [52].

Resolution rule: Locked

EXAMPLE 3: `<SmilBaseSet>SMIL-3GPP-R6</SmilBaseSet>`

Attribute name: **SmilModules**

Attribute definition: This attribute defines a list of SMIL 2.0 modules supported by the client. If the SmilBaseSet is used those modules do not need to be explicitly listed here. In that case only additional module support needs to be listed.

Component: Streaming

Type: Literal (Bag)

Legal values: SMIL 2.0 module names defined in the SMIL 2.0 recommendation [31], section 2.3.3, table 2.

Resolution rule: Append

EXAMPLE 4:
```
<SmilModules>
  <rdf:Bag>
    <rdf:li>BasicTransitions</rdf:li>
    <rdf:li>MulitArcTiming</rdf:li>
  </rdf:Bag>
</SmilModules>
```

5.2.3.3 Attributes from UAProf

In the UAProf vocabulary [40] there are several attributes that are of interest for the PSS. The formal definition of these attributes is given in [40]. The following list of attributes is recommended for PSS applications:

Attribute name: **BitsPerPixel**

Component: HardwarePlatform

Attribute description: The number of bits of colour or greyscale information per pixel

EXAMPLE 1: `<BitsPerPixel>8</BitsPerPixel>`

Attribute name: **ColorCapable**

Component: HardwarePlatform

Attribute description: Whether the device display supports colour or not.

EXAMPLE 2: `<ColorCapable>Yes</ColorCapable>`

Attribute name: **PixelAspectRatio**

Component: HardwarePlatform

Attribute description: Ratio of pixel width to pixel height

EXAMPLE 3:

```
<PixelAspectRatio>1x2</PixelAspectRatio>
```

Attribute name: **PointingResolution**

Component: HardwarePlatform

Attribute description: Type of resolution of the pointing accessory supported by the device.

EXAMPLE 4:

```
<PointingResolution>Pixel</PointingResolution>
```

Attribute name: **Model**

Component: HardwarePlatform

Attribute description: Model number assigned to the terminal device by the vendor or manufacturer

EXAMPLE 5:

```
<Model>Lexus</Model>
```

Attribute name: **Vendor**

Component: HardwarePlatform

Attribute description: Name of the vendor manufacturing the terminal device

EXAMPLE 6:

```
<Vendor>Toyota</Vendor>
```

Attribute name: **CcppAccept-Charset**

Component: SoftwarePlatform

Attribute description: List of character sets the device supports

EXAMPLE 7:

```
<CcppAccept-Charset>
  <rdf:Bag>
    <rdf:li>UTF-8</rdf:li>
  </rdf:Bag>
</CcppAccept-Charset>
```

Attribute name: **CcppAccept-Encoding**

Component: SoftwarePlatform

Attribute description: List of transfer encodings the device supports

EXAMPLE 8:

```
<CcppAccept-Encoding>
  <rdf:Bag>
    <rdf:li>base64</rdf:li>
  </rdf:Bag>
</CcppAccept-Encoding>
```

Attribute name: **CcppAccept-Language**

Component: SoftwarePlatform

Attribute description: List of preferred document languages

EXAMPLE 9:

```
<CcppAccept-Language>
  <rdf:Seq>
    <rdf:li>en</rdf:li>
    <rdf:li>se</rdf:li>
  </rdf:Seq>
</CcppAccept-Language>
```

5.2.4 Extensions to the PSS schema/vocabulary

5.2.4.1 Vocabulary definitions

The use of RDF enables an extensibility mechanism for CC/PP-based schemas that addresses the evolution of new types of devices and applications. The Release-6 PSS profile schema specification has been updated from Release 5 and has thus been assigned a unique RDF schema. ~~The PSS profile schema specification is going to provide a base vocabulary but i~~The following URIs uniquely identify the RDF schemas for Release 5 and Release 6:

PSS Release 5 URI: http://www.3gpp.org/profiles/PSS/ccppschema-PSS5

PSS Release 6 URI: http://www.3gpp.org/profiles/PSS/ccppschema-PSS6

In the future new usage scenarios might have need for expressing new attributes. If the base vocabulary is further updated, a new unique namespace will be assigned to the updated schema. The base vocabulary shall only be changed by the TSG responsible for the present document. All extensions to the profile schema shall be governed by the rules defined in [40] clause 7.7.

5.2.4.2 Backward compatibility

An important issue when introducing a new vocabulary is to ensure backward compatibility. PSS Release-6 clients should seamlessly work together with PSS Release-5 servers and vice versa. To obtain backward compatibility, a Release-6 client should provide servers with multiple device-capability profiles using PSS Release-5 and Release-6 vocabularies, respectively. This can be done by providing two URIs referring to two separate profiles or one URI referring to one combined profile that uses both the Relase-5 and the Release-6 namespaces. PSS Release-6 servers should handle both namespaces, whereas PSS Release-5 servers will ignore profiles with unknown namespaces.

5.2.5 Signalling of profile information between client and server

When a PSS client or server support capability exchange it shall support the profile information transport over both HTTP and RTSP between client and server as defined in clause 9.1 (including its subsections) of the WAP 2.0 UAProf specification [40] with the following additions:

- The "x-wap-profile" and "x-wap-profile-diff" headers may not be present in all HTTP or RTSP request. That is, the requirement to send this header in all requests has been relaxed.
- The defined headers may be applied to both RTSP and HTTP.
- The "x-wap-profile-diff" header is only valid for the current request. The reason is that PSS does not have the WSP session concept of WAP.
- Push is not relevant for the PSS.

The following recommendations are made to how and when profile information should be sent between client and server:

- PSS content servers supporting capability exchange shall be able to receive profile information in all HTTP and RTSP requests.
- The terminal should not send the "x-wap-profile-diff" header over the air-interface since there is no compression scheme defined.
- RTSP: the client should send profile information in the DESCRIBE message. It may send it in any other request.

If the terminal has some prior knowledge about the file type it is about to retrieve, e.g. file extensions, the following apply:

- HTTP and SDP: when retrieving an SDP with HTTP the client should include profile information in the GET request. This way the HTTP server can deliver an optimised SDP to the client.
- HTTP and SMIL: When retrieving a SMIL file with HTTP the client should include profile information in the GET request. This way the HTTP server can deliver an optimised SMIL presentation to the client. A SMIL presentation can include links to static media. The server should optimise the SMIL file so that links to the referenced static media are adapted to the requesting client. When the "x-wap-profile-warning" indicates that content selection has been applied (201-203) the PSS client should assume that no more capability exchange has to be performed for the static media components. In this case it should not send any profile information when retrieving static media to be included in the SMIL presentation. This will minimise the HTTP header overhead.

5.2.6 Merging device capability profiles

Profiles need to be merged whenever the PSS server receives multiple device capability profiles. Multiple occurrences of attributes and default values make it necessary to resolve the profiles according to a resolution process.

The resolution process shall be the same as defined in UAProf [40] clause 6.4.1.

- Resolve all indirect references by retrieving URI references contained within the profile.
- Resolve each profile and profile-diff document by first applying attribute values contained in the default URI references and by second applying overriding attribute values contained within the category blocks of that profile or profile-diff.
- Determine the final value of the attributes by applying the resolved attribute values from each profile and profile-diff in order, with the attribute values determined by the resolution rules provided in the schema. Where no resolution rules are provided for a particular attribute in the schema, values provided in profiles or profile-diffs are assumed to override values provided in previous profiles or profile-diffs.

When several URLs are defined in the "x-wap-profile" header and there exists any attribute that occurs more than once in these profiles the rule is that the attribute value in the second URL overrides, or is overridden by, or is appended to the attribute value from the first URL (according to the resolution rule) and so forth. This is what is meant with "Determine the final value of the attributes by applying the resolved attribute values from each profile and profile-diff in order, with…" in the third bullet above. If the profile is completely or partly inaccessible or otherwise corrupted the server should still provide content to the client. The server is responsible for delivering content optimised for the client based on the received profile in a best effort manner.

NOTE: For the reasons explained in Annex A clause A.4.3 the usage of indirect references in profiles (using the CC/PP defaults element) is not recommended.

5.2.7 Profile transfer between the PSS server and the device profile server

The device capability profiles are stored on a device profile server and referenced with URLs. According to the profile resolution process in clause 5.2.6 of the present document, the PSS server ends up with a number of URLs referring to profiles and these shall be retrieved.

- The device profile server shall support HTTP 1.1 for the transfer of device capability profiles to the PSS server.
- If the PSS server supports capability exchange it shall support HTTP 1.1 for transfer of device capability profiles from the device profile server. A URL shall be used to identify a device capability profile.

- Normal content caching provisions as defined by HTTP apply.

5.3 Session set-up and control

5.3.1 General

Continuous media is media that has an intrinsic time line. Discrete media on the other hand does not itself contain an element of time. In this specification speech, audio and video belongs to first category and still images and text to the latter one.

Streaming of continuous media using RTP/UDP/IP (see clause 6.2) requires a session control protocol to set-up and control of the individual media streams. For the transport of discrete media (images and text), vector graphics, timed text and synthetic audio this specification adopts the use of HTTP/TCP/IP (see clause 6.3). In this case there is no need for a separate session set-up and control protocol since this is built into HTTP. This clause describes session set-up and control of the continuous media speech, audio and video.

5.3.2 RTSP

RTSP [5] shall be used for session set-up and session control. PSS clients and servers shall follow the rules for minimal on-demand playback RTSP implementations in appendix D of [5]. In addition to this:

- PSS servers and clients shall implement the DESCRIBE method (see clause 10.2 in [5]);
- PSS servers and clients shall implement the Range header field (see clause 12.29 in [5]);
- PSS servers shall include the Range header field in all PLAY responses;
- PSS servers and clients should implement the SET_PARAMETER method (see clause 10.9 in [5]);
- PSS servers and clients should implement the Bandwidth header field (see clause 12.6 in [5];
- PSS servers and clients should implement the 3GPP-Link-Char header field (see clause 5.3.2.1);
- PSS servers and clients should implement the 3GPP-Adaptation header field (see clause 5.3.2.2).

5.3.2.1 The 3GPP-Link-Char header

To enable PSS clients to report the link characteristics of the radio interface to the PSS server, the "3GPP-Link-Char" RTSP header is defined. The header takes one or more arguments. The reported information should be taken from a QoS reservation (i.e. the QoS profile as defined in [56]). Note that this information is only valid for the wireless link and does not apply end-to-end. However, the parameters do provide constraints that can be used.

Three parameters are defined that can be included in the header, and future extensions are possible to define. Any unknown parameter shall be ignored. The three parameters are:

- "GBW": the link's guaranteed bit-rate in kilobits per second as defined by [56];
- "MBW": the link's maximum bit-rate in kilobits per second as defined by [56];
- "MTD": the link's maximum transfer delay, as defined by [56] in milliseconds.

The "3GPP-Link-Char" header syntax is defined below using ABNF [53]:

```
3gpplinkheader      = "3GPP-Link-Char" ":" link-char-spec *("," 0*1SP link-char-spec) CRLF

link-char-spec      = char-link-url *(";" 0*1SP link-parameters)

char-link-url       = "url" "=" <">url<">

link-parameters     = Guaranteed-BW / Max-BW / Max-Transfer-delay / extension-type

Guaranteed-BW       = "GBW" "=" 1*DIGIT  ; bps

Max-BW              = "MBW" "=" 1*DIGIT ; bps

Max-Transfer-delay  = "MTD" "=" 1*DIGIT  ; ms
```

```
extension-type      = token "=" (token / quoted-string)

DIGIT               = as defined in RFC 2326 [5]

token               = as defined in RFC 2326 [5]

quoted-string       = as defined in RFC 2326 [5]

url                 = as defined in RFC 2326 [5]
```

The "3GPP-Link-Char" header can be included in a request using any of the following RTSP methods: SETUP, PLAY, OPTIONS, and SET_PARAMETER. The header shall not be included in any response. The header can contain one or more characteristics specifications. Each specification contains a URI that can either be an absolute or a relative, any relative URI use the RTSP request URI as base. The URI points out the media component that the given parameters apply to. This can either be an individual media stream or a session aggregate.

If a QoS reservation (PDP context) is shared by several media components in a session the 3GPP-Link-Char header shall not be sent prior to the RTSP PLAY request. In this case the URI to use is the aggregated RTSP URI. If the QoS reservation is not shared (one PDP context per media) the media stream URI must be used in the 3GPP-Link-Char specification. If one QoS reservation (PDP context) per media component is used, the specification parameters shall be sent per media component.

The "3GPP-Link-Char" header should be included in a SETUP or PLAY request by the client, to give the initial values for the link characteristics. A SET_PARAMETER or OPTIONS request can be used to update the 3GPP-Link-Char values in a session currently playing. It is strongly recommended that SET_PARAMETER is used, as this has the correct semantics for the operation and also requires less overhead both in bandwidth and server processing. When performing updates of the parameters, all of the previous signalled values are undefined and only the given ones in the update are defined. This means that even if a parameter has not changed, it must be included in the update.

Example:

```
3GPP-LinkChar: url="rtsp://server.example.com/media.3gp"; GBW=32; MBW=128; MTD=2000
```

In the above example the header tells the server that its radio link has a QoS setting with a guaranteed bit-rate of 32 kbps, a maximum bit-rate of 128 kbps, and a maximum transfer delay of 2.0 seconds. These parameters are valid for the aggregate of all media components, as the URI is an aggregated RTSP URI.

5.3.2.2 The 3GPP-Adaptation header

To enable PSS clients to set bit-rate adaptation parameters, a new RTSP request and response header is defined. The header can be used in the methods SETUP, PLAY, OPTIONS, and SET_PARAMETER. The header defined in ABNF [53] has the following syntax:

```
3GPP-adaptation-def = "3GPP-Adaptation" ":" adaptation-spec 0*("," adaptation-spec)

adaptation-spec     = url-def *adapt-params

adapt-params        = ";" buffer-size-def

                    / ";" target-time-def

url-def             = "url" "=" <"> url <">

buffer-size-def     = "size" "=" 1*9DIGIT ; bytes

target-time-def     = "target-time" "=" 1*9DIGIT; ms

url                 = ( absoluteURI / relativeURI )
```

absoluteURI and relativeURI are defined in RFC 2396 [60] and updated in RFC 2732 [61]. The base URI for any relative URI is the RTSP request URI.

The "3GPP-Adaptation" header shall be sent in responses to requests containing this header. The PSS server shall not change the values in the response header. The presence of the header in the response indicates to the client that the server acknowledges the request.

The buffer size signalled in the "3GPP-Adaptation" header shall correspond to a reception and de-jittering buffer that has this given amount of space for complete RTP packets including the RTP header. The specified buffer size shall also include any Annex G pre-decoder buffer space used for this media, as the two buffers cannot be separated.

The target protection time signalled in the value of the "target-time" parameter is the targeted minimum buffer level or, in other words, the client desired amount of playback time in milliseconds to guarantee interrupt-free playback and allow the server to adjust the transmission rate, if needed.

5.3.2.3 The Quality of Experience headers

5.3.2.3.1 Protocol initiation and termination

A new RTSP header is defined to enable the PSS client and server to negotiate which Quality of Experience (QoE) metrics the PSS client should send, how often they should be sent and how to turn the metrics transmission off. This header can be present in requests and responses of RTSP methods SETUP, SET_PARAMETER, OPTIONS (with Session ID) and PLAY. The header is defined in ABNF [53] as follows (see [53] for specifiers not defined here):

```
QoE-Header       = "3GPP-QoE-Metrics" ":" ("Off" / Measure-Spec *("," Measure-Spec)) CRLF

Measure-Spec     = Stream-URL";" ((Metrics ";" Sending-rate [";" Measure-Range] *([";" Parameter_Ext])) /
                   “Off”)

Stream-URL       = "url" "="  <”>Rtsp_URL<”>

Metrics          = "metrics" "=" "{"Metrics-Name *("," Metrics-Name) " }"

Metrics-Name     =  1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7c / 0x7e) ;VCHAR except “;”, “,”, “{“ or ”}”

Sending-Rate     = “rate" "=" 1*DIGIT / "End"

Measure-Range    = "range" "=" Ranges-Specifier

Parameter_Ext    = “On”/”Off”/ (1*DIGIT [“.” 1*DIGIT]) / (1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7c /
                   0x7e))

Ranges-Specifier = as defined in RFC 2326 [5]

Rtsp_URL         = as defined in RFC 2326 [5]
```

There are two ways to use this header:

- Using only the "Off" parameter is an indication that either server or client wants to cancel the metrics reporting.
- Using other parameters indicates a request to start the metrics transmission.

If "Stream-URL" is an RTSP Session Control URL, then "Metrics" applies to the RTSP session. If "Stream-URL" is an RTSP Media Control URL, then "Metrics" apply only to the indicated media component of the session.

QoE metrics with the same "Stream-URL", "Sending-rate" and "Measure-Range" shall be aggregated within a single "Measure-Spec" declaration. Otherwise, multiple "Stream-URL" declarations shall be used.

The "Metrics" field contains the list of names that describes the metrics/measurements that are required to be reported in a PSS session. The names that are not included in the "Metrics" field shall not be reported during the session.

The "Sending-Rate" shall be set, and it expresses the maximum time period in seconds between two successive QoE reports. If the "Sending-Rate" value is 0, then the client shall decide the sending time of the reports depending on the events occurred in the client. Values ≥ 1 indicate a precise reporting interval. The shortest interval is one second and the longest interval is undefined. The reporting interval can be different for different media, but it is recommended to maintain a degree of synchronization in order to avoid extra traffic in the uplink direction. The value "End" indicates that only one report is sent at the end of the session.

The optional "Measure-Range" field, if used, shall define the time range in the stream for which the QoE metrics will be reported. There shall be only one range per measurement specification. The range format shall be any of the formats allowed by the media. If the "Measure-Range" field is not present, the corresponding (media or session level) range attribute in SDP shall be used. If SDP information is not present, the metrics range shall be the whole session duration.

There shall be only one "3GPP-QoE-Metrics" header in one RTSP request or response.

5.3.2.3.2 Metrics feedback

The QoE metrics feedback can be conveyed in requests to the PSS server using the SET_PARAMETER, PAUSE or TEARDOWN methods by the "3GPP-QoE-Feedback" header. The header is defined in ABNF [53] as follows (see [53] for specifiers not defined here):

```
Feedbackheader  = "3GPP-QoE-Feedback" ":" Feedback-Spec *("," Feedback-Spec) CRLF

Feedback-Spec   = Stream-URL 1*(";" Parameters) [";" Measure-Range]

Stream-URL      = as specified in clause 5.3.2.3.1

Parameters      =  Metrics-Name "=" "{" SP / (Measure *(","Measure)) "}"

Metrics-Name    = as defined in clause 5.3.2.3.1

Measure         = Value [SP Timestamp]

Measure-Range   = as defined in clause 5.3.2.3.1

Value           = (1*DIGIT ["." *DIGIT]) /  1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7c / 0x7e)
                  ;VCHAR except “;”, “,”, “{“ or ”}”

Timestamp       =  NPT-Time

NPT-Time        = as defined in RFC 2326 [5]
```

"Stream-URL" is the RTSP session or media control URL that identifies the media the feedback parameter applies to.

The "Metrics-Name" field in the "Parameters" definition contains the name of the metrics/measurements and uses the same identifiers as the "3GPP-QoE-Metrics" header in clause 5.3.2.3.1.

The "Value" field indicates the results. There is the possibility that the same event occurs more than once during a monitoring period. In that case the metrics value may occur more than once indicating the number of events to the server.

The optional "Timestamp" (defined in NPT time) indicates the time when the event occurred or when the metric was calculated. If no events have occurred, it shall be reported with an empty set (only containing a space).

The optional "Measure-Range" indicates the actual reporting period, for which this report is valid.

QoE metrics reporting should be done by the PSS client by using the SET_PARAMETER method. However, for more efficiency, RTSP PAUSE and TEARDOWN methods may also be used in particular cases, such as:

CASE 1: When sending the very last QoE report, the client should embed the QoE information into a TEARDOWN message.

CASE 2: When the client wants to pause the streaming flow, QoE information should be embedded into a PAUSE method. The PSS client should not send any QoE reports to the PSS server when the system is paused, since there is no media flow.

5.3.3 SDP

5.3.3.1 General

RTSP requires a presentation description. SDP shall be used as the format of the presentation description for both PSS clients and servers. PSS servers shall provide and clients interpret the SDP syntax according to the SDP specification [6] and appendix C of [5]. The SDP delivered to the PSS client shall declare the media types to be used in the session using a codec specific MIME media type for each media. MIME media types to be used in the SDP file are described in clause 5.4 of the present document.

The SDP [6] specification requires certain fields to always be included in an SDP file. Apart from this a PSS server shall always include the following fields in the SDP:

- "a=control:" according to clauses C.1.1, C.2 and C.3 in [5];
- "a=range:" according to clause C.1.5 in [5];
- "a=rtpmap:" according to clause 6 in [6];
- "a=fmtp:" according to clause 6 in [6].

When an SDP document is generated for media stored in a 3GP file, each control URL defined at the media-level “a=control:” field shall include a stream identifier in the last segment of the path component of the URL. The value of the stream id shall be defined by the track-ID field in the track header (tkhd) ~~atom~~ box associated with the media track. When a PSS server receives a set-up request for a stream, it shall use the stream identifier specified in the URL to map the request to a media track with a matching track-ID field in the 3GP file. Stream identifiers shall be expressed using the following syntax:

streamIdentifier = <stream_id_token>"~~“~~="~~”~~<stream_id>

stream_id_token = 1*alpha

stream_id = 1*digit

The bandwidth field in SDP is needed by the client in order to properly set up QoS parameters. Therefore, a PSS server shall include the “b=AS:” field at the media level for each media stream in SDP, and a PSS client shall interpret this field. When a PSS client receives SDP, it should ignore the session level “b=AS:” parameter (if present), and instead calculate session bandwidth from the media level bandwidth values of the relevant streams. A PSS client shall also handle the case where the bandwidth parameter is not present, since this may occur when connecting to a Release-4 server.

Note that for RTP based applications , ‘b=AS:’ gives the RTP "session bandwidth" (including UDP/IP overhead) as defined in section 6.2 of [9].

The bandwidth for RTCP traffic shall be described using the "RS" and "RR" SDP bandwidth modifiers, as specified by [5~~2~~]. The "RS" SDP bandwidth modifier indicates the RTCP bandwidth allocated to the sender (i.e. PSS server) and "RR" indicates the RTCP bandwidth allocated to the receiver (i.e. PSS client). A PSS server shall include the "b=RS:" and "b=RR:" fields at the media level for each media stream in SDP, and a PSS client shall interpret them. A PSS client shall also handle the case where the bandwidth modifier is not present according to section 3 of [5~~2~~], since this may occur when connecting to a Release-4 server.

There shall be a limit on the allowed RTCP bandwidth for senders and receivers in a session. This limit is defined as follows:

~~□~~- 4000 bps for the RS field (at media level);

~~□~~- 5000 bps for the RR field (at media level).

The default value for each of the "RS" and "RR" SDP bandwidth modifiers is 2.5% of the bandwidth given by the “b=AS” parameter at media level.

In Annex A.2.1 an example SDP in which the limit for the total RTCP bandwidth is 5% of the session bandwidth is presented.

The media which has an SDP description that include an open ended range (format=startvalue-) in any time format in the SDP attribute "a=range", e.g. "a=range: npt=now-", or "a=range: clock=20030825T152300Z-", shall be considered media of unknown length. Such a media shall be considered as non-seekable, unless other attributes override this property.

The "t=", "r=", and "z=" SDP parameters are used to indicate when the described session is active. It can be used for users to filter out obsolete SDP files. When creating an SDP for a streaming session, one should try to come up with the most accurate estimate of time that the session is active. The "t=", "r=", and "z=" SDP parameters are used for this purpose, i.e., to indicate when the described session is active. If the time at which a session is active is known to be only for a limited period, the "t=", "r=", and "z=" attributes should be filled out appropriately (the "t=" should contain non-zero values, possibly using the "r=" and "z=" parameters). If the stop-time is set to zero, the session is not bounded, though it will not become active until after the start-time. If the start-time is also zero, the session is regarded as permanent. A session should only be marked as permanent ("t=0 0") if the session is going to be available for a significantly long period of time or if the start and stop times are not known at the time of SDP file creation. Recommendations for what is considered a significant time is present in the SDP specification [6].

IPv6 addresses in SDP descriptions shall be supported according to RFC 3266[49].

NOTE: The SDP parsers and/or interpreters shall be able to accept NULL values in the 'c=' field (e.g. 0.0.0.0 in IPv4 case). This may happen when the media content does not have a fixed destination address. For more details, see Section C.1.7 of [5] and Section 6 of [6].

5.3.3.2 Additional SDP fields

The following Annex G-related media level SDP fields are defined for PSS:

- "a=X-predecbufsize:<size of the hypothetical pre-decoder buffer>"
  If rate adaptation (see clause 10.2) is not in use, ~~T~~this gives the suggested size of the Annex G hypothetical pre-decoder buffer in bytes.

  If rate adaptation is in use, this gives the suggested minimum size of a buffer (hereinafter called the pre-decoder buffer) that is used to smooth out transmit time variation (compared to flat-bitrate transmission scheduling) and video bitrate variation.

- "a=X-initpredecbufperiod:<initial pre-decoder buffering period>"
  If rate adaptation is not in use, t~~T~~his gives the required initial pre-decoder buffering period specified according to Annex G. Values are interpreted as clock ticks of a 90-kHz clock. That is, the value is incremented by one for each 1/90 000 seconds. For example, value 180 000 corresponds to a two second initial pre-decoder buffering.

  If rate adaptation is in use, this gives the suggested minimum greatest difference in RTP timestamps in the pre-decoder buffer after any de-interleaving has been applied. Note that X-initpredecbufperiod is expressed as clock ticks of a 90-kHz clock. Hence, conversion may be required if the RTP timestamp clock frequency is not 90 kHz.

- "a=X-initpostdecbufperiod:<initial post-decoder buffering period>"
  If rate adaptation is not in use, t~~T~~his gives the required initial post-decoder buffering period specified according to Annex G. Values are interpreted as clock ticks of a 90-kHz clock.

  If rate adaptation is in use, this gives the initial post-decoder buffering period assuming that the hypothetical decoding and post-decoder buffering model given in points 5 to 10 in Annex G clause G.3 would be followed. Note that the operation of the post-decoder buffer is logically independent from rate adaptation and is used to compensate non-instantaneous decoding of pictures.

- "a=X-decbyterate:<peak decoding byte rate>"
  This gives the peak decoding byte rate that was used to verify the compatibility of the stream with Annex G. Values are given in bytes per second.

If none of the attributes "a=X-predecbufsize:", "a=X-initpredecbufperiod:", "a=X-initpostdecbufperiod:", and "a=x-decbyterate:" is present, clients should not expect a packet stream according to Annex G. If at least one of the listed attributes is present, and if the client does not choose the usage of bit-rate adaptation via RTSP as described in clause 5.3.2.2, the transmitted video packet stream shall conform to Annex G. If at least one of the listed attributes is present, but some of the listed attributes are missing in an SDP description, clients should expect a default value for the missing attributes according to Annex G.

The following media level SDP field is defined for PSS:

- "a=framesize:<payload type number> <width>-<height>"
  This gives the largest video frame size of H.263 streams.

The frame size field in SDP is needed by the client in order to properly allocate frame buffer memory. For MPEG-4 visual streams, the frame size shall be extracted from the "~~“~~config"~~”~~ information in the SDP. For H.263 streams, a PSS server shall include the "~~“~~a=framesize"~~”~~ field at the media level for each stream in SDP, and a PSS client should interpret this field, if present. Clients should be ready to receive SDP descriptions without this attribute.

If this attribute is present, the frame size parameters shall exactly match the largest frame size defined in the video stream. The width and height values shall be expressed in pixels.

5.3.3.3 The “alt” and “alt-default-id” attributes

The client should interpret the following two media level attributes: "alt" and "alt-default-id". A client from earlier releases will ignore these attributes and can safely do so in a correctly formatted SDP. If the attributes are used by the server they shall be used in a way that makes them backward compatible. When interpreted, they define a number of alternatives from which the client can select the most appropriate one.

A non-extended SDP gives only one alternative for each media part (Annex A.1 Example 1). This is the default alternative for each media. The new SDP attributes defined here are used to modify the default attributes or to add new attributes to the default attributes thus creating new alternatives. Each alternative is numerically identified.

The alternative attribute "alt" is used to replace or add an SDP line to the default configuration. If the alternative attribute contains an SDP line, for which the type and the modifier already exist in the default alternative, the default must be replaced with the given line(s). In case there are multiple lines with the same type and modifier in the default alternative, all of the lines must be replaced. Multiple alternative lines can be used to modify the default alternative. The alternative lines that are used to form a certain alternative shall all carry the same numerical identifier (Annex A.1, Examples 2-4).

The alternative identifier is a unique identifier that points out a single alternative in one media declaration. The identifier must be unique between all media descriptions and their alternatives as it is used for creating combinations between different medias with the grouping attribute (see 5.3.3.4).

The default configuration is in itself a valid alternative. Therefore an attribute (alt-default-id) is defined that assigns an alternative identifier to the default alternative. This identifier can then be used with the grouping attribute (see 5.3.3.4) to create combinations of alternatives from different medias.

The alternative attribute is defined below in BNF from RFC 2234 [53]. The SDP line is any SDP line allowed at media level except "m=".

alt   = "a" "=" "alt" ":" alt-id ":" SDP-line CRLF

SDP-line   = <type>=<value> ; See RFC 2327

alt-id   = 1*DIGIT ; unique identifier for the alternative in whole SDP.

To be able to assign an alternative ID to the default alternative, the following identification attribute is defined.

alt-default-id   = "a" "=" "alt-default-id" ":" alt-id CRLF

5.3.3.4 The session level grouping attribute, “alt-group”

The client should handle the following attribute: "alt-group". A client from earlier releases will ignore this attribute and can safely do so. When interpreted, it defines a number of grouping alternatives from which the client can select the most appropriate one. The identifiers defined in 5.3.3.3 are used together with the "alt-group" attribute to create combinations consisting of, e.g., one audio and one video alternative. It is the server’s responsibility to create meaningful grouping alternatives.

A grouping attribute is used to recommend certain combinations of media alternatives to the client. There may be more than one grouping attribute at the session level as long as they are for different grouping types and subtypes.

alt-group = "a" "=" "alt-group" ":" alt-group-type ":" alt-group-subtype ":" alt-grouping *(";" alt-grouping) CRLF

alt-group-type   = token   ; "token" defined in RFC 2327 [6]

alt-group-subtype   = token

alt-grouping   = grouping-value "=" alt-id *("," alt-id)

grouping-value   = token

The alt-group attribute gives one or more combinations of alternatives through their IDs. Each grouping must be given a grouping value. The grouping value is used to determine if the alternatives within the grouping suits the client. New types and subtypes can be added later.

The following grouping types and subtypes are defined:

- Type: BW, Subtype: All modifiers defined for the SDP "b=" attribute at session and media level. See www.IANA.org for current list of registered attributes.

  Grouping value: The bandwidth value defined for that modifier calculated over all the alternatives grouped together in that grouping. For SDP bandwidth modifiers defined at session level the value shall be calculated according to its rule over the alternative part of the grouping. For media-level-only modifiers, the grouping value shall be calculated as a sum of the media-level values in the grouped alternatives. Note: The meaning of a sum may not be clearly defined but should give a decent enough indication for the grouping.

  Grouping recommendations: Each grouping should only contain one alternative from each media type. There is no need to give groupings for all combinations between the media alternatives, rather it is strongly recommended to only give the most suitable combinations (Annex A.1 Example 5). The client can use the bandwidth values of the grouping to estimate the minimum, guaranteed or maximum bandwidth that will be needed for that session.

- Type: LANG Subtype: RFC3066

  Grouping value: A language tag as defined by RFC 3066 [54]. The grouping MUST contain all media alternatives, which support that language tag.

  Grouping recommendations: It is recommended that other mechanisms, like user profiles if existing, are primarily used to ensure that the content has language suitable for the user (Annex A.1, Example 6).

Se also Annex A1, Examples 7 through 16. In the examples all three new attributes "alt", "alt-default-id" and "alt-group" are used.

5.3.3.5 The bit-rate adaptation support attribute, “3GPP-Adaptation-Support”

To signal the support of bit-rate adaptation, a media level only SDP attribute is defined in ABNF [53]:

sdp-Adaptation-line = "a" "=" "3GPP-Adaptation-Support" ":" report-frequency CRLF

report-frequency = 1*2DIGIT

A server implementing rate adaptation shall signal the "3GPP-Adaptation-Support" attribute in its SDP.

A client receiving an SDP description where the SDP attribute "3GPP-Adaptation-Support" is present knows that the server provides rate adaptation. The client, if it supports bit-rate adaptation, shall then in its subsequent RTSP signalling use the “3GPP-Adaptation” header as defined in clause 5.3.2.2, as well as the RTCP OBSN APP packet for reporting of the oldest buffered sequence number, as defined in clause 6.2.3.2.

The SDP attribute shall only be present at the media level. The report frequency value indicates to the client that it shall include an OBSN APP packet in at least every "report-frequency" compound RTCP packet. For example, if this value is 3, the client shall send the OBSN APP packet in at least every 3rd RTCP packet.

5.3.3.6 The Quality of Experience support attribute, "3GPP-QoE-Metrics"

SDP can be used to initiate the QoE negotiation. The reason why SDP is needed is to support the use cases where SDP is distributed through other methods than RTSP DESCRIBE, e.g. WAP, HTTP or email. A new SDP attribute, which can be used either at session or media level, is defined below in ABNF [53] based on RFC 2327 [6]:

QoE-Metrics-line = "a" "=" "3GPP-QoE-Metrics:" att_measure_spec *("," att-measure-spec)) CRLF

att-measure-spec = Metrics ";" Sending-rate [";" Measure-Range] *([";" Parameter_Ext])

Metrics = as defined in clause 5.3.2.3.1.

Sending-Rate = as defined in clause 5.3.2.3.1.

```
Measure-Range        = as defined in clause 5.3.2.3.1.

Parameter_Ext        = as defined in clause 5.3.2.3.1.
```

A server uses this attribute to indicate that QoE metrics are supported and shall be used if also supported by the client. When present at session level, it shall only contain metrics that apply to the complete session. When present at media level, it shall only contain metrics that are applicable to individual media. The URI that is used in the specification of the RTSP header "3GPP-QoE-Metrics:" is implicit by the RTSP control URI (a=control).

5.3.3.7 The asset information attribute, "3GPP-Asset-Information"

This asset information attribute is defined to transmit asset information in SDP. The attribute is defined ABNF [53]:

```
3GPP-Assets-Info     = "a" "=" "3GPP-Asset-Information:" Asset 0*("," Asset) CRLF

Asset                = ("{" "url" "=" <">URL<"> "}") / ("{"AssetName "=" AssetBox "}")

URL                  = as defined in [60]

AssetName            = "Title" / "Description" / "Copyright" / "Performer" / "Author" / "Genre" / "Rating" /

                        "Classification" / "Keywords" / "Location" / asset-extension

asset-extension      = 1*((0x01..0x09) / 0x0b / 0x0c / (0x0e..0x1f) / (0x21..0x2b) / (0x2d..0x3c) / (0x3e..0x7a) /

                        0x7c / (0x7d..0xff))   ;any byte except SP, NUL, CR, LF , "=", ",", "{" or "}"

AssetBox             = Base64 encoded version [69] of any asset box as defined in Clause 8 of [50].
```

This SDP attribute can be present at session level, media level or both. Multiple instances of the attribute are allowed.

The resource referenced by the URL can be any pre-formatted data, e.g. an XHTML page or XML file, containing any asset information. It is up to the client’s capability and user’s preference to render the information pointed by the URL.

Example 17 in Clause A.1 shows an SDP file that includes the "3GPP-Asset-Information" attribute.

5.4 MIME media types

For continuous media (speech, audio and video) the following MIME media types shall be used:

- AMR narrow-band speech codec (see clause 7.2) MIME media type as defined in [11];
- AMR wideband speech codec (see clause 7.2) MIME media type as defined in [11];
- MPEG-4 AAC audio codec (see clause 7.3) MIME media type as defined in RFC 3016 [13]. When used in SDP the attribute “cpresent” SHALL be set to “0” indicating that the configuration information is only carried out of band in the SDP “config” parameter;
- MPEG-4 video codec (see clause 7.4) MIME media type as defined in RFC 3016 [13]. When used in SDP the configuration information shall be carried outband in the "config" SDP parameter and inband (as stated in RFC 3016). As described in RFC 3016, the configuration information sent inband and the config information in the SDP shall be the same except that first_half_vbv_occupancy and latter_half_vbv_occupancy which, if exist, may vary in the configuration information sent inband;
- H.263 [22] video codec (see clause 7.4) MIME media type as defined in ~~annex C, clause C.1 of the present document.~~clause 4.2.7 of [62].

MIME media types for JPEG, GIF, PNG, SP-MIDI, Mobile DLS, Mobile XMF, SVG, timed text and XHTML can be used both in the "Content-type" field in HTTP and in the "type" attribute in SMIL 2.0. The following MIME media types shall be used for these media:

- JPEG (see clause 7.5) MIME media type as defined in [15];
- GIF (see clause 7.6) MIME media type as defined in [15];

- PNG (see sub clause 7.6) MIME media type as defined in [38];
- SP-MIDI (see sub clause 7.3A) MIME media type as defined in clause C.2 in Annex C of the present document;
- DLS MIME media type to represent Mobile DLS (see sub clause 7.3A) as defined in clause C.4 in Annex C of the present document;
- Mobile XMF (see sub clause 7.3A) MIME media type as defined in clause C.3 in Annex C of the present document;
- SVG (see sub clause 7.7) MIME media type as defined in [42];
- XHTML (see clause 7.8) MIME media type as defined in [16];
- Timed text (see subclause 7.9) MIME media type as defined in~~ clause D.9 in Annex D of the present document~~ [50].

MIME media type used for SMIL files shall be according to [31] and for SDP files according to [6].

6 Data transport

6.1 Packet based network interface

PSS clients and servers shall support an IP-based network interface for the transport of session control and media data. Control and media data are sent using TCP/IP [8] and UDP/IP [7]. An overview of the protocol stack can be found in figure 2 of the present document.

6.2 RTP over UDP/IP

6.2.1 General

The IETF RTP [9] ~~and [10]~~ provides means for sending real-time or streaming data over UDP (see [7]). The encoded media is encapsulated in the RTP packets with media specific RTP payload formats. RTP payload formats are defined by IETF. RTP also provides a protocol called RTCP (see clause 6 in [9]) for feedback about the transmission quality. ~~For the calculation of the RTCP transmission interval Annex A.7 in [9] shall be used. Clause A.3.2.3 in Annex A of the present document provides more information about the minimum RTCP transmission interval.~~

RTP/UDP/IP transport of continuous media (speech~~,~~, audio and video) shall be supported. Sending of RTCP shall be performed according to the used RTP profile, indicated RTCP bandwidth, and other RTCP related parameters. The transmission times of RTCP shall be controlled by algorithms performing as the ones specified in the RTP specification [9], and if AVPF is used according to [57]. For information on how the RTCP transmission interval depends on different values of the RTCP parameters, see Annex A.3.2.3.

6.2.2 RTP profiles

For RTP/UDP/IP transport of continuous media the following RTP profile shall be supported:

- RTP Profile for Audio and Video Conferences with Minimal Control [10], also called RTP/AVP;

For RTP/UDP/IP transport of continuous media the following RTP profile should be supported:

- Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [57], also called RTP/AVPF. A PSS client or server is not required to support the feedback formats specified in section 6 of [57], however the RTCP packet type defined shall at least be possible to ignore.

Clause A.3.2.3 in Annex A of the present document provides more information about the minimum RTCP transmission interval.

6.2.3 RTCP extensions

6.2.3.1 RTCP extended reports

A PSS client should implement the framework and SDP signalling of the RTP Control Protocol Extended Reports [58]. A PSS client should further implement the following report formats:

- Loss RLE Report Block defined in section 4.1of [58].

A PSS client should send the report block(s) indicated by SDP signalling from the PSS server. A PSS server may limit the report blocks size using SDP signalling. For best utility the client should report in every packet and provide redundancy by reporting also on past RTCP intervals. In cases where the size restriction prevents the client from reporting on all the RTP packets, the client shall first remove the redundant reporting. Only if this action is not enough to reduce the reports to satisfactory sizes, should thinning be applied.

6.2.3.2 RTCP App packet for client buffer feedback (OBSN APP packet)

To report the oldest buffered sequence number (OBSN) for bit-rate adaptation, an RTCP APP packet is defined. The format of a generic RTCP APP packet is shown in Figure 3 below:

```
 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| subtype |   PT=APP=204  |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          name (ASCII)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   application-dependent data                ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Generic Format of an RTCP APP packet.

For rate adaptation the name and subtype fields must be set to the following values:

name: The OBSN APP data format is detected through the name "PSS0", i.e. 0x50535330 and the subtype.
subtype: This field shall be set to 0 for the OBSN format.
length: The number of 32 bit words –1, as defined in RFC 3550 [9]. This means that the field will be 2+2*N, where N is the number of sources reported on. The length field will typically be 4, i.e. 20 bytes packets.
application-dependent data: One or more of the following data format blocks (as described in Figure 4) can be included in the application-dependent data location of the APP packet. The APP packets length field is used to detect how many blocks of data are present. The block shall be sent for the SSRCs for which there is a report block, part of either a Receiver Report or a Sender Report, included in the RTCP compound packet. An OBSN APP packet shall not contain any other data format than the one described in figure 4 below.

```
 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
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              SSRC                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Playout Delay            |             OBSN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Data format block for OBSN reporting

SSRC: The SSRC of the media stream the buffered packets belong to.

OBSN: Oldest Buffered Sequence Number. The RTP sequence number of the oldest packet present in the announced buffer space for the SSRC reported on. In other words, it is the sequence number of the first packet in the sequence of packets to be played out. In the cases the buffer does not contain any packets for this SSRC, the next not yet received sequence number shall be reported, i.e. an OBSN value that is one larger than the least significant 16 bits of the RTCP SR or RR report block's "extended highest sequence number received".

Playout delay: The difference between the scheduled playout time of the oldest packet and the time of sending the OBSN APP packet in milliseconds. The client may choose not to indicate this value by using the reserved value (Ox FFFF). In case of an empty buffer, the playout delay is not defined and the client should also use the reserved value 0xFFFF for this field.
The playout delay allows the server to have a more precise value of the amount of time before the client will underflow. The playout delay shall be computed until the actual media playout (i.e., audio playback or video display).

6.2.4 RTP payload formats

For RTP/UDP/IP transport of continuous media the following RTP payload formats shall be used:

- AMR narrow-band speech codec (see clause 7.2) RTP payload format according to [11]. A PSS client is not required to support multi-channel sessions;
- AMR wideband speech codec (see clause 7.2) RTP payload format according to [11]. A PSS client is not required to support multi-channel sessions;
- MPEG-4 AAC audio codec (see clause 7.3) RTP payload format according to RFC 3016 [13];
- MPEG-4 video codec (see clause 7.4) RTP payload format according to RFC 3016 [13];
- H.263 video codec (see clause 7.4) RTP payload format according to RFC 2429 [14].

NOTE: The payload format RFC 3016 for MPEG-4 AAC specify that the audio streams shall be formatted by the LATM (Low-overhead MPEG-4 Audio Transport Multiplex) tool [21]. It should be noted that the references for the LATM format in the RFC 3016 [13] point to an older version of the LATM format than included in [21]. In [21] a corrigendum to the LATM tool is included. This corrigendum includes changes to the LATM format making implementations using the corrigendum incompatible with implementations not using it. To avoid future interoperability problems, implementations of PSS client and servers supporting AAC shall follow the changes to the LATM format included in [21].

6.3 HTTP over TCP/IP

The IETF TCP provides reliable transport of data over IP networks, but with no delay guarantees. It is the preferred way for sending the scene description, text, bitmap graphics and still images. There is also need for an application protocol to control the transfer. The IETF HTTP [17] provides this functionality.

HTTP/TCP/IP transport shall be supported for:

- still images (see clause 7.5);
- bitmap graphics (see clause 7.6);

- synthetic audio (see clause 7.3A);
- vector graphics (see clause 7.7);
- text (see clause 7.8);
- timed text (see clause 7.9);
- scene description (see clause 8);
- presentation description (see clause 5.3.3).

HTTP/TCP/IP transport should be supported for:

- 3GP files for progressive download (see clause 7.10).

6.4 Transport of RTSP

Transport of RTSP shall be supported according to RFC 2326 [5].

7 Codecs

7.1 General

For PSS offering a particular media type, media decoders are specified in the following clauses.

7.2 Speech

If speech is supported, ~~T~~the AMR decoder shall be supported for narrow-band speech [18][63][64][65]. The AMR wideband speech decoder, [20][66][67][68], shall be supported when wideband speech working at 16 kHz sampling frequency is supported.

7.3 Audio

If audio is supported, MPEG-4 AAC Low Complexity (AAC-LC) object type decoder [21] should be supported. The maximum sampling rate to be supported by the decoder is 48 kHz. The channel configurations to be supported are mono (1/0) and stereo (2/0). In addition, the MPEG-4 AAC Long Term Prediction (AAC-LTP) object type decoder may be supported.

When a server offers an AAC-LC or AAC-LTP stream with the specified restrictions, it shall include the "profile-level-id" and "object" MIME parameters in the SDP "a=fmtp" line. The following values shall be used:

Object Type	profile-level-id	object
AAC-LC	15	2
AAC-LTP	15	4

7.3a Synthetic audio

If synthetic audio is supported, t~~T~~he Scalable Polyphony MIDI (SP-MIDI) content format defined in Scalable Polyphony MIDI Specification [44] and the device requirements defined in Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP [45] should be supported.

SP-MIDI content is delivered in the structure specified in Standard MIDI Files 1.0 [46], either in format 0 or format 1.

In addition the Mobile DLS instrument format defined in [70] and the Mobile XMF content format defined in [71] should be supported.

A PSS client supporting Mobile DLS shall meet the minimum device requirements defined in [70] in section 1.3 and the requirements for the common part of the synthesizer voice as defined in [70] in sections 1.2.1.2. If Mobile DLS is supported, wavetables encoded with the G.711 A-law codec (wFormatTag value 0x0006, as defined in [70]) shall also be supported. The optional group of processing blocks as defined in [70] may be supported. Mobile DLS resources are delivered either in the file format defined in [70], or within Mobile XMF as defined in [71]. For Mobile DLS files delivered outside of Mobile XMF, the loading application should unload Mobile DLS instruments so that the sound bank required by the SP-MIDI profile [45] is not persistently altered by temporary loadings of Mobile DLS files.

Content that pairs Mobile DLS and SP-MIDI resources is delivered in the structure specified in Mobile XMF [71]. As defined in [71], a Mobile XMF file shall contain one SP-MIDI SMF file and no more than one Mobile DLS file. PSS clients supporting Mobile XMF must not support any other resource types in the Mobile XMF file. Media handling behaviours for the SP-MIDI SMF and Mobile DLS resources contained within Mobile XMF are defined in [71].

7.4 Video

If video is supported, ITU-T Recommendation H.263 [22] profile 0 level 10 decoder shall be supported. ~~This is the mandatory video decoder for the PSS.~~ In addition, a PSS client should support:

- H.263 [23] Profile 3 Level 10 decoder;
- MPEG-4 Visual Simple Profile Level 0 decoder, [24] and [25].

~~These two video decoders are optional to implement.~~

~~An optional~~ The video buffer model ~~is~~ given in Annex G of the present document should be supported if video is supported.

NOTE: ITU-T Recommendation H.263 profile 0 has been mandated to ensure that video-enabled PSS supports a minimum baseline video capability. Both H.263 and MPEG-4 visual decoders can decode an H.263 profile 0 bitstream. It is strongly recommended, though, that an H.263 profile 0 bitstream is transported and stored as H.263 and not as MPEG-4 visual (short header), as MPEG-4 visual is not mandated by PSS.

7.5 Still images

If still images are supported, ISO/IEC JPEG [26] together with JFIF [27] decoders shall be supported. The support for ISO/IEC JPEG only applies~~y~~ to the following two modes:

- baseline DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF0' in [26];
- progressive DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF2' [26].

7.6 Bitmap graphics

If bitmap graphics is supported, ~~T~~the following bitmap graphics decoders should be supported:

- GIF87a, [32];
- GIF89a, [33];
- PNG, [38].

7.7 Vector graphics

If vector graphics is supported, ~~T~~the SVG Tiny profile [42] [43] shall be supported. In addition SVG Basic profile [42] [43] may be supported.

NOTE: The compression format for SVG content is GZIP [59], in accordance with the SVG specification [42].

7.8 Text

The text decoder is intended to enable formatted text in a SMIL presentation.

If text is supported, a~~A~~ PSS client shall support

- text formatted according to XHTML Mobile Profile [47];
- rendering a SMIL presentation where text is referenced with the SMIL 2.0 "text" element together with the SMIL 2.0 "src" attribute.

If text is supported, t~~T~~he following character coding formats shall be supported:

- UTF-8, [30];

~~-~~- UCS-2, [29].

NOTE: Since both SMIL and XHTML are XML based languages it would be possible to define a SMIL plus XHTML profile. In contrast to the presently defined ~~PSS4~~ SMIL Language Profile that only contain SMIL modules, such a profile would also contain XHTML modules. No combined SMIL and XHTML profile is specified for PSS. Rendering of such documents is out of the scope of the present document.

7.9 Timed text

If timed text is supported, PSS clients shall support [51]~~Annex D, clause D.8a, of this specification~~ with 3GP files using Basic profile [50]. There is no support for RTP transport of timed text in this release; 3GP files containing timed text may only be downloaded.

NOTE: When a PSS client supports timed text it needs to be able to receive and parse 3GP files containing the text streams. This does not imply a requirement on PSS clients to be able to render other continuous media types contained in 3GP files, e.g. AMR and H.263, if such media types are included in a presentation together with timed text. Audio and video are instead streamed to the client using RTSP/RTP (see clause 6.2).

7.10 3GPP file format

3GP files [50] can be used by both PSS clients and PSS servers. The following profiles are used:

- Basic profile shall be supported by PSS clients if timed text is supported;
- Progressive-download profile should be supported by PSS clients;
- Streaming server profile should be supported by PSS servers.

8 Scene description

8.1 General

The 3GPP PSS uses a subset of SMIL 2.0 [31] as format of the scene description. PSS clients and servers with support for scene descriptions shall support the 3GPP ~~PSS~~ SMIL Language Profile defined in [52]~~clause 8.2 (abbreviated 3GPP PSS SMIL)~~. This profile is a subset of the SMIL 2.0 Language Profile, but a superset of the SMIL 2.0 Basic Language Profile. ~~The present d~~Document [52] also includes an informative Annex A~~B~~ that provides guidelines for SMIL content authors.

NOTE: The interpretation of this is not that all streaming sessions are required to use SMIL. For some types of sessions, e.g. consisting of one single continuous media or two media synchronised by using RTP timestamps, SMIL may not be needed.

~~8.2 3GPP PSS SMIL Language Profile~~

~~8.2.1 Introduction~~

~~3GPP PSS SMIL is a markup language based on SMIL Basic [31] and SMIL Scalability Framework.~~

~~3GPP PSS SMIL consists of the modules required by SMIL Basic Profile (and SMIL 2.0 Host Language Conformance) and additional MediaAccessibility, MediaDescription, MediaClipping, MetaInformation, PrefetchControl, EventTiming and BasicTransitions modules. All of the following modules are included:~~

- ~~SMIL 2.0 Content Control Modules -- BasicContentControl, SkipContentControl and PrefetchControl~~
- ~~SMIL 2.0 Layout Module -- BasicLayout~~
- ~~SMIL 2.0 Linking Module – BasicLinking, LinkingAttributes~~
- ~~SMIL 2.0 Media Object Modules -- BasicMedia, MediaClipping, MediaAccessibility and MediaDescription~~
- ~~SMIL 2.0 Metainformation Module -- Metainformation~~
- ~~SMIL 2.0 Structure Module -- Structure~~
- ~~SMIL 2.0 Timing and Synchronization Modules -- BasicInlineTiming, MinMaxTiming, BasicTimeContainers, RepeatTiming and EventTiming~~
- ~~SMIL 2.0 Transition Effects Module -- BasicTransitions~~

~~8.2.2 Document Conformance~~

~~A conforming 3GPP PSS SMIL document shall be a conforming SMIL 2.0 document.~~

~~All 3GPP PSS SMIL documents use SMIL 2.0 namespace.~~

~~<smil xmlns="`http://www.w3.org/2001/SMIL20/Language`">~~

~~3GPP PSS SMIL documents may declare requirements using systemRequired attribute:~~

~~EXAMPLE 1: <smil xmlns="`http://www.w3.org/2001/SMIL20/Language`"~~
~~xmlns:EventTiming="http://www.w3.org/2000/SMIL20/CR/EventTiming "~~
~~systemRequired="EventTiming">~~

~~Namespace URI http://www.3gpp.org/SMIL20/PSS5/ identifies the version of the 3GPP PSS SMIL profile described in the present document. Authors may use this URI to indicate requirement for exact 3GPP PSS SMIL semantics for a document or a subpart of a document:~~

~~EXAMPLE 2: <smil xmlns="`http://www.w3.org/2001/SMIL20/Language`"~~
~~xmlns:pss5="http://www.3gpp.org/SMIL20/PSS5/"~~
~~systemRequired="pss5">~~

~~The content authors should generally not include the PSS requirement in the document unless the SMIL document relies on PSS specific semantics that are not part of the W3C SMIL. The reason for this is that SMIL players that are not conforming 3GPP PSS user agents may not recognize the PSS URI and thus refuse to play the document.~~

~~8.2.3 User Agent Conformance~~

~~A conforming 3GPP PSS SMIL user agent shall be a conforming SMIL Basic User Agent.~~

~~A conforming user agent shall implement the semantics 3GPP PSS SMIL as described in clauses 8.2.4 and 8.2.5 (including subclauses).~~

~~A conforming user agent shall recognise~~

- ~~the URIs of all included SMIL 2.0 modules;~~

- ~~the URI http://www.3gpp.org/SMIL20/PSS5/ as referring to all modules and semantics of the version of the 3GPP PSS SMIL profile described in the present document;~~
- ~~the URI http://www.3gpp.org/SMIL20/PSS4/ as referring to all modules and semantics of the 3GPP PSS SMIL profile defined in Release 4 of the present document.~~

~~NOTE: The difference between PSS4 and PSS5 is that the BasicTransitions module has been added in PSS5.~~

~~8.2.4 3GPP PSS SMIL Language Profile definition~~

~~3GPP PSS SMIL is based on SMIL 2.0 Basic language profile [31]. This chapter defines the content model and integration semantics of the included modules where they differ from those defined by SMIL Basic.~~

~~8.2.4.1 Content Control Modules~~

~~3GPP PSS SMIL includes the content control functionality of the BasicContentControl, SkipContentControl and PrefetchControl modules of SMIL 2.0. PrefetchControl is not part of SMIL Basic and is an additional module in this profile.~~

~~All BasicContentControl attributes listed in the module specification shall be supported.~~

> ~~NOTE: The SMIL specification [31] defines that all functionality of PrefetchControl module is optional. This mean that even although PrefetchControl is mandatory user agents may implement semantics of PrefetchControl module only partially or not to implement them at all.~~

~~PrefetchControl module adds the **prefetch** element to the content model of SMIL Basic **body, switch, par** and **seq** elements. The **prefetch** element has the attributes defined by the PrefetchControl module (**mediaSize, mediaTime and bandwidth**), the **src** attribute, the BasicContentControl attributes and the **skip-content** attribute.~~

~~8.2.4.2 Layout Module~~

~~3GPP PSS SMIL includes the BasicLayout module of SMIL 2.0 for spatial layout. The module is part of SMIL Basic.~~

~~Default values of the width and height attributes for root-layout shall be the dimensions of the device display area.~~

~~8.2.4.3 Linking Module~~

~~3GPP PSS SMIL includes the SMIL 2.0 BasicLinking and LinkingAttributes modules for providing hyperlinks between documents and document fragments. The BasicLinking module is from SMIL Basic.~~

~~When linking to destinations outside the current document, implementations may ignore values "play" and "pause" of the 'sourcePlaystate' attribute and values "new" and "pause" of the 'show' attribute, instead using the semantics of values "stop" and "replace" respectively. When the values of 'sourcePlaystate' and 'show' are ignored the player may also ignore the 'sourceLevel' attribute since it is of no use then~~

~~8.2.4.4 Media Object Modules~~

~~3GPP PSS SMIL includes the media elements from the SMIL 2.0 BasicMedia module and attributes from the MediaAccessibility, MediaDescription and MediaClipping modules. MediaAccessibility, MediaDescription and MediaClipping modules are additions in this profile to the SMIL Basic.~~

~~See clause 5.4 for what are the mandatory and optional MIME types a 3GPP PSS SMIL player needs to support.~~

~~MediaClipping module adds to the profile the ability to address sub-clips of continuous media. MediaClipping module adds '**clipBegin**' and '**clipEnd**´(and for compatibility '**clip-begin**' and '**clip-end**') attributes to all media elements.~~

~~MediaAccessibility module provides basic accessibility support for media elements. New attributes '**alt**', '**longdesc**' and '**readIndex**' are added to all media elements by this module. MediaDescription module is included by the MediaAccessibility module and adds '**abstract**', '**author**' and '**copyright**' attributes to media elements.~~

~~8.2.4.5 Metainformation Module~~

~~The MetaInformation module of SMIL 2.0 is included to the profile. This module is addition in this profile to the SMIL Basic and provides a way to include descriptive information about the document content into the document.~~

~~This module adds **meta** and **metadata** elements to the content model of SMIL Basic **head** element.~~

~~8.2.4.6 Structure Module~~

~~The Structure module defines the top-level structure of the document. It is included by SMIL Basic.~~

~~8.2.4.7 Timing and Synchronization modules~~

~~The timing modules included in the 3GPP SMIL are BasicInlineTiming, MinMaxTiming, BasicTimeContainers, RepeatTiming and EventTiming. The EventTiming module is an addition in this profile to the SMIL Basic.~~

~~For 'begin' and 'end' attributes either single offset-value or single event-value shall be allowed. Offsets shall not be supported with event-values.~~

~~Event timing attributes that reference invalid IDs (for example elements that have been removed by the content control) shall be treated as being indefinite.~~

~~Supported event names and semantics shall be as defined by the SMIL 2.0 Language Profile. All user agents shall be able to raise the following event types:~~

- ~~activateEvent;~~
- ~~beginEvent;~~
- ~~endEvent.~~

~~The following SMIL 2.0 Language event types should be supported:~~

- ~~focusInEvent;~~
- ~~focusOutEvent;~~
- ~~inBoundsEvent;~~
- ~~outOfBoundsEvent;~~
- ~~repeatEvent.~~

~~User agents shall ignore unknown event types and not treat them as errors.~~

~~Events do not bubble and shall be delivered to the associated media or timed elements only.~~

~~8.2.4.8 Transition Effects Module~~

~~3GPP PSS SMIL profile includes the SMIL 2.0 BasicTransitions module to provide a framework for describing transitions between media elements.~~

> ~~NOTE: The SMIL specification [31] defines that all functionality of BasicTransitions module is optional: "Transitions are hints to the presentation. Implementations must be able to ignore transitions if they so desire and still play the media of the presentation". This mean that even although the BasicTransitions module is mandatory user agents may implement semantics of the BasicTransitions module only partially or not to implement them at all. Content authors should use transitions in their SMIL presentation where this appears useful. User agents that fully support the semantics of the Basic Transitions module will render the presentation with the specified transitions. All other user agents will leave out the transitions but present the media content correctly.~~

~~User agents that implement the semantics of this module should implement at least the following transition effects described in SMIL 2.0 specification [31]:~~

- ~~barWipe;~~

- ~~irisWipe;~~
- ~~clockWipe;~~
- ~~snakeWipe;~~
- ~~pushWipe;~~
- ~~slideWipe;~~
- ~~fade;~~

~~A user agent should implement the default subtype of these transition effects.~~

~~A user agent that implements the semantics of this module shall at least support transition effects for non-animated image media elements. For purposes of the Transition Effects modules, two media elements are considered overlapping when they occupy the same region.~~

~~BasicTransitions module adds attributes 'transIn' and 'transOut' to the media elements of the Media Objects modules, and value "transition" to the set of legal values for the 'fill' attribute of the media elements. It also adds transition element to the content model of the head element.~~

~~8.2.5 Content Model~~

~~This table shows the full content model and attributes of the 3GPP PSS SMIL profile. The attribute collections used are defined by SMIL Basic ([31], SMIL Host Language Conformance requirements, chapter 2.4). Changes to SMIL Basic are shown in **bold**.~~

~~Table 1: Content model for the 3GPP PSS SMIL profile~~

~~**Element**~~		
	~~**Elements**~~	~~**Attributes**~~
~~smil~~	~~head, body~~	~~COMMON-ATTRS, CONTCTRL-ATTRS, xmlns~~
~~head~~	~~layout, switch, **meta, metadata, transition**~~	~~COMMON-ATTRS~~
~~body~~	~~TIMING-ELMS, MEDIA-ELMS, switch, a, **prefetch**~~	~~COMMON-ATTRS~~
~~layout~~	~~root-layout, region~~	~~COMMON-ATTRS, CONTCTRL-ATTRS, type~~
~~root-layout~~	~~EMPTY~~	~~COMMON-ATTRS, backgroundColor, height, width, skip-content~~
~~region~~	~~EMPTY~~	~~COMMON-ATTRS, backgroundColor, bottom, fit, height, left, right, showBackground, top, width, z-index, skip-content, regionName~~
~~ref, animation, audio, img, video, text, textstream~~	~~area~~	~~COMMON-ATTRS, CONTCTRL-ATTRS, TIMING-ATTRS, repeat, region, MEDIA-ATTRS, **clipBegin(clip-begin), clipEnd(clip-end), alt, longDesc, readIndex, abstract, author, copyright, transIn, transOut**~~
~~a~~	~~MEDIA-ELMS~~	~~COMMON-ATTRS, LINKING-ATTRS~~
~~area~~	~~EMPTY~~	~~COMMON-ATTRS, LINKING-ATTRS, TIMING-ATTRS, repeat, shape, coords, nohref~~
~~par, seq~~	~~TIMING-ELMS, MEDIA-ELMS, switch, a, **prefetch**~~	~~COMMON-ATTRS, CONTCTRL-ATTRS, TIMING-ATTRS, repeat~~
~~switch~~	~~TIMING-ELMS, MEDIA-ELMS, layout, a, **prefetch**~~	~~COMMON-ATTRS, CONTCTRL-ATTRS~~
~~**prefetch**~~	~~**EMPTY**~~	~~**COMMON-ATTRS, CONTCTRL-ATTRS, mediaSize, mediaTime, bandwidth, src, skip-content**~~
~~**meta**~~	~~**EMPTY**~~	~~**COMMON-ATTRS, content, name, skip-content**~~
~~**metadata**~~	~~**EMPTY**~~	~~**COMMON-ATTRS, skip-content**~~
~~**transition**~~	~~**EMPTY**~~	~~**COMMON-ATTRS, CONTCTRL-ATTRS, type, subtype, startProgress, endProgress, direction, fadeColor. skip-content, dur**~~

9 3GPP file format (interchange format for MMS)

The 3GPP file format is defined in [50].

~~9.1 General~~

~~The 3GPP file format (3GP) is based on the ISO base media file format [50] and is defined in this specification. It is mandated in [35] to be used for continuous media along the entire delivery chain envisaged by the MMS, independent on whether the final delivery is done by streaming or download, thus enhancing interoperability.~~

~~In particular, the following stages are considered:~~

- ~~upload from the originating terminal to the MMS proxy;~~
- ~~file exchange between MMS servers;~~
- ~~transfer of the media content to the receiving terminal, either by file download or by streaming. In the first case the self-contained file is transferred, whereas in the second case the content is extracted from the file and streamed according to open payload formats. In this case, no trace of the file format remains in the content that goes on the wire/in the air.~~

~~Additionally, the 3GPP file format should be used for the storage in the servers and the "hint track" mechanism may be used for the preparation for streaming.~~

~~Clause 9.2 of the present document gives the necessary requirements to follow for the 3GPP file format used in MMS. These requirements will guarantee PSS to interwork with MMS as well as the 3GPP file format to be used internally within the MMS system. For PSS servers not interworking with MMS there is no requirement to follow these guidelines.~~

~~9.2 3GPP file format conformance~~

~~The 3GPP file format, used in this specification for timed multimedia (such as video, associated audio and timed text), is structurally based on the ISO base media file format defined in [50]. However, the conformance statement for 3GP files is defined in the present document by addressing the registration of codecs, file identification, file extension and MIME type definition.~~

~~NOTE: Codecs or functionalities not conforming to a 3GP file may be ignored.~~

~~9.2.1 Registration of non-ISO codecs~~

~~MPEG-4 video and AAC audio code streams, as well as the non-ISO code streams AMR narrow-band speech, AMR wideband speech, H.263 encoded video and timed text can be included in 3GP files as described in annex D of the present document.~~

~~9.2.2 Hint tracks~~

~~Hint tracks are a mechanism that a server implementation may choose to use in preparation for the streaming of media content contained in 3GP files. However, it should be observed that the usage of hint tracks is an internal implementation matter for the server, and it falls outside the scope of the present document.~~

~~9.2.3 Limitations to the ISO base media file format~~

~~The following limitations to the ISO base media file format [50] shall apply to a 3GP file of this specification:~~

- ~~there shall be no references to external media outside the file, i.e. a 3GP file shall be self-contained;~~
- ~~the maximum number of tracks shall be one for video, one for audio and one for text;~~
- ~~the maximum number of sample entries shall be one per track for video and audio (but unrestricted for text);~~

- ~~compact sample sizes ('stz2') shall not be used;~~
- ~~movie fragments shall not be used.~~

~~NOTE: If a file contains video or audio tracks with more than one sample entry per track, a reader may skip those tracks or the entire file.~~

~~9.2.4 MPEG-4 systems specific elements~~

~~For the storage of MPEG-4 media specific information in 3GP files, this specification refers to MP4 [51], which is also based on the ISO base media file format. However, tracks relative to MPEG-4 system architectural elements (e.g. BIFS scene description tracks or OD Object descriptors) are optional in 3GP files and shall be ignored. The inclusion of MPEG-4 media does not imply the usage of MPEG-4 systems architecture. The receiving terminal is not required to implement any of the specific MPEG-4 system architectural elements.~~

~~9.2.5 Interpretation of 3GPP file format~~

~~All index numbers used in the 3GPP file format start with the value one rather than zero, in particular "first-chunk" in Sample to chunk box, "sample-number" in Sync sample box and "shadowed-sample-number", "sync-sample-number" in Shadow sync sample box.~~

~~9.2.6 3GPP file authoring guidelines for progressive download~~

~~The present document specifies the 3GPP file format to be used for distribution of continuous media clips using MMS. The same file format can also be used for file download from any server. However, to achieve a better response to a user request for media, it is often advantageous if the client can start playing the media before the full file is downloaded. This scenario is known as progressive download and is provided by many proprietary media solutions. It is the purpose of this clause to point out that this is also easily achievable by using the 3GPP file format. This possibility has been inherent in the file format from the first version in Release 4, and the only thing that is needed is that the content creator follows the guidelines provided here.~~

~~The principles behind progressive download are that the session information should be put at the beginning of the file and that the media tracks should be interleaved within the file. In practice, this leads to the following guidelines for the creation of 3GP files:~~

- ~~the 'moov' box should be placed at the start of the file, right after the 'ftyp' box;~~
- ~~the media tracks should be interleaved inside the file. The typical interleaving length is one second.~~

~~For the release-4 file format, the boxes are called atoms but, except for that, everything applies equally well.~~

~~It should be noted that no change is needed at the server side, and that a client that does not support progressive download can always play the file once it has been completely downloaded. A progressive download client can start playing a 3GP file that has been created along the progressive download guidelines once it has received a first chunk of all media in the session. If the file has not been prepared for progressive download, the client will always need to wait for the full download.~~

10 Adaptation of continuous media

10.1 General

The PSS includes a number of protocols and functionalities that can be utilized to allow the PSS session to adapt transmission and content rates to the available network resources. The goal of this is of course to achieve highest possible quality of experience for the end-user with the available resources, while maintaining interrupt-free playback of the media. This requires that the available network resources are estimated and that transmission rates are adapted to the available network link rates. This can prevent overflowing network buffers and thereby avoid packet losses. The real-time properties of the transmitted media must be considered so that media does not arrive too late to be useful. This will require that media content rate is adapted to the transmission rate.

To avoid buffer overflows, resulting in that the client must discard useful data, while still allowing the server to deliver as much data as possible into the client buffer, a functionality for client buffer feedback is defined. This allows the server to closely monitor the buffering situation on the client side and to do what it is capable in order to avoid client buffer underflow. The client specifies how much buffer space the server can utilize and the desired target level of protection. When the desired level of protection is achieved, the server may utilize any resources beyond what is needed to maintain that protection level to increase the quality of the media. The server can also utilize the buffer feedback information to decide if the media quality needs to be lowered in order to avoid a buffer underflow and the resulting play-back interruption.

10.2 Bit-rate adaptation

The bit-rate adaptation for PSS is server centric in the meaning that transmission and content rate are controlled by the server. The server use RTCP and RTSP as the basic information sources about the state of the client and network. This allows link-rate adaptation also when communicating with PSS clients of earlier releases, as long as they send RTCP receiver reports frequently enough.

10.2.1 Link-rate estimation

The actual algorithm providing the link-rate estimation is implementation specific. However, this chapter describes and gives rules for the different information sources that can be used for link-rate estimation.

10.2.1.1 Initial values

A PSS client should inform the server the quality of service parameters for the used wireless link. The known parameters should be included in the RTSP "3GPP-Link-Char" header (chapter 5.3.2.1) in either the RTSP SETUP or PLAY request. This enables the server to set some basic assumption about the possible bit-rates and link response. If the client has initially reported these parameters and they are changed during the session the client shall update these parameters by including the "3GPP-Link-Char" header in a SET_PARAMETER or OPTIONS request.

A PSS client should inform the server about initial bit-rate available over the link, if known. This reporting shall be done using the RTSP "Bandwidth" header in either the RTSP SETUP or PLAY request. The QoS negotiated guaranteed bit-rate is the best estimate for the bandwidth value.

10.2.1.2 Regular information sources

The basic information source giving regular reports useful for bit-rate estimations is the RTCP receiver reports as defined by [9]. The RTCP reporting interval is dependent on the RTP profile in use, the bit-rate assigned to RTCP, the average size of RTCP packets, and the number of reporting entities. Most of these parameters can be set or affected by the PSS server through signalling. This allows the server to configure the reporting interval to a desirable working point. See chapter 5.3.3.1 for specification on how the RTCP bandwidth is signalled by the server.

In most PSS RTP sessions the server and the client only have one SSRC each, thus providing the highest possible reporting rate. However some scenarios could result in that the number of used SSRC is larger, thereby possibly lowering the effective reporting interval for client, server or both.

The average size of the RTCP packets cannot be tightly controlled, but a loose control is possible by controlling which RTCP packet types that are used. This will depend on which of the below-listed RTCP extensions are in use.

The PSS server can signal the PSS client in SDP, to request that "Loss RLE Report Block" in RTCP XR (section 6.2.3) are used to report packet loss vectors.

10.2.2 Transmission adaptation

The transmission adaptation is implementation dependent. The 3GPP file format server extensions [50] provide a server the possibility to store alternative encodings useful for stream switching.

A server doing transmission rate adaptation through content rate adaptation shall still deliver content according to the SDP description of the media streams, e.g. a video stream delivered after content rate adaptation must still belong to the SDP announced profile and be consistent with any configuration. This will either put restrictions on the possible alternatives or require declaration of several RTP payload types or media encodings that might not be used.

10.2.3 Signalling for client buffer feedback

The client buffer feedback signalling functionality should be supported by PSS clients and PSS servers. For PSS clients and servers that support the client buffer feedback signalling functionality, the following parts shall be implemented:

- SDP service support, as described in clause 5.3.3.5.
- The size (in bytes) of the buffer the client provides for rate adaptation. It is signalled to the server through RTSP, as described in clause 5.3.2.2
- The target buffer protection time (in milliseconds). It is signalled to the server through RTSP, as described in clause 5.3.2.2.
- The sequence number of the oldest ("oldest buffered sequence number") packet in the client buffer. It is signalled to the server via RTCP, as described in clause 6.2.3.2.

If a PSS server supports client buffer feedback, it shall include the attribute "3GPP-Adaptation-Support" in the SDP, as described in clause 5.3.3.5. Upon reception of such an SDP attribute, if a PSS client supports client buffer feedback, it shall in the SETUP for each individual media include the "3GPP-Adaptation" header. Furthermore, upon reception of a successful SETUP response (including "3GPP-Adaptation" header), the PSS client shall send OBSN APP packets according to clause 5.3.3.5.

The "3GPP-Adaptation" header may be included in PLAY, OPTIONS and SET_PARAMETER requests in order to update the target buffer protection time value during a session. The buffer size value shall not be modified during a session.

With the buffer size, the oldest buffered sequence number parameters, and by means of the "Highest Received Sequence Number" already contained in RTCP receiver reports, the server can calculate the number of bytes in the client buffer at the sending time of the last received RTCP report. Based on the calculated client buffer fill level, the server can avoid overflowing the buffer. This level will also allow the server to detect when the buffer level drops and thus react to try to prevent underflow. The time before the client buffer will underflow can be estimated by the server by referring to the timestamp of the packet of highest sequence number, the timestamp of the packet of oldest sequence number and the playout delay of the packet of oldest sequence number, if signalled. The playout delay improves the accuracy of the estimated time before the client underflows. For example, in the case of low frame-rate video, the playout delay may contribute significantly to the total buffering time at the client.

The level of protection needed against transmission rate variations over a wireless network can be substantial (throughput variation because of network load, radio conditions, several seconds of interruption because of handovers, possible extra buffering to perform retransmission). In order to minimise the initial buffering delay, the client may choose an initial buffering that is less than the required buffering it has determined would be satisfactory. For this reason, the target buffer protection time indicates the amount of playable media (in time), which the client would like to have in its buffer. Therefore a server should not perform content adaptation towards higher content rates until the given target time of media units is available in the buffer.

11 Quality of Experience

11.1 General

The PSS Quality of Experience (QoE) metrics feature is optional for both PSS servers and clients, and shall not disturb the PSS service. A PSS server that supports the QoE metrics feature shall signal the activation and gathering of client QoE metrics when desired. A 3GPP PSS client supporting the feature shall perform the quality measurements in accordance to the measurement definitions, aggregate them into client QoE metrics and report the metrics to the PSS server using the QoE transport protocol when so requested. The way the QoE metrics are processed and made available is out of the scope of this specification.

11.2 QoE metrics

A PSS client should measure the metrics at the transport layer, but may also do it at the application layer for better accuracy.

The reporting period for the metrics is the period over which a set of metrics is calculated. The maximum value of the reporting period is negotiated via the QoE protocol as in clause 11.3. The reporting period shall not include any voluntary event that impacts the actual play, such as pause or rewind, or any buffering or freezes/gaps caused by them.

The following metrics shall be derived by the PSS client implementing QoE. All the metrics defined below are only applicable to at least one of audio, video, speech and timed text media types, and are not applicable to other media types such as synthetic audio, still images, bitmap graphics, vector graphics, and text. Any unknown metrics shall be ignored by the client and not included in any QoE report.

11.2.1 Corruption duration metric

Corruption duration, M, is the time period from the NPT time of the last good frame before the corruption, to the NPT time of the first subsequent good frame or the end of the reporting period (whichever is sooner). A corrupted frame may either be an entirely lost frame, or a media frame that has quality degradation and the decoded frame is not the same as in error-free decoding. A good frame is a "completely received" frame X that

- either it is a refresh frame (does not reference any previously decoded frames AND where none of the subsequent received frames reference any frames decoded prior to X);
- or does not reference any previously decoded frames;
- or references previously decoded "good frames".

"Completely received" means that all the bits are received and no bit error has occurred.

Corruption duration, M, in milliseconds can be calculated as below:

a) M can be derived by the client using the codec layer, in which case the codec layer signals the decoding of a good frame to the client. A good frame could also be derived by error tracking methods, but decoding quality evaluation methods shall not be used.

b) In the absence of information from the codec layer, M should be derived from the NPT time of the last frame before the corruption and N, where N is optionally signalled from server to client and represents the maximum duration between two subsequent refresh frames in milliseconds.

c) In the absence of information from the codec layer and if N is not signalled, then M defaults to ∞ (for video) or to one frame duration (for audio), or the end of the reporting period (whichever is sooner).

The optional parameter N as defined in point b is used with the "Corruption_Duration" parameter in the "3GPP-QoE-Metrics" header. Another optional parameter T is defined to indicate whether the client uses error tracking or not. The value of T shall be set by the client. The syntax for N and T to be included in the "Measure-Spec" (clause 5.3.2.3.1) is as follows:

N = "N" "=" 1*DIGIT

T = "T" "=" "On" / "Off"

The syntax for the "Metrics-Name Corruption_Duration" for the QoE-Feedback header is as defined in clause 5.3.2.3.2

The absence of an event can be reported using the space (SP).

For the "Metrics-Name Corruption_Duration", the "Value" field in 5.3.2.3.2 indicates the corruption duration. The unit of this metrics is expressed in milliseconds. There is the possibility that corruption occurs more than once during a reporting period. In that case the value can occur more than once indicating the number of corruption events.

The value of "Timestamp" is equal to the NPT time of the last good frame inside the reporting period, in playback order, before the occurrence of the corruption, relative to the starting time of the reporting period. If there is no good frame inside the reporting period and before the corruption, the timestamp is set to the starting time of the reporting period.

11.2.2 Rebuffering duration metric

Rebuffering is defined as any stall in playback time due to any involuntary event at the client side.

The syntax for the “Metrics-Name Rebuffering_Duration" for the QoE-Feedback header is as defined in clause 5.3.2.3.2.

The absence of an event can be reported using the space (SP).

For the "Metrics-Name Rebuffering_Duration", the "Value" field in 5.3.2.3.2 indicates the rebuffering duration. The unit of this metrics is expressed in seconds, and can be a fractional value. There is the possibility that rebuffering occurs more than once during a reporting period. In that case the metrics value can occur more than once indicating the number of rebuffering events.

The optional "Timestamp" indicates the time when the rebuffering has occurred since the beginning of the reporting period. The value of the "Timestamp" is equal to the NPT time of the last played frame inside the reporting period and before the occurrence of the rebuffering, relative to the starting time of the reporting period. If there is no played frame inside the reporting period, the timestamp is set to the starting time of the reporting period.

11.2.3 Initial buffering duration metric

Initial buffering duration is the time from receiving the first RTP packet until playing starts.

The syntax for the "Metrics-Name Initial_Buffering_Duration" for the QoE-Feedback header is as defined in clause 5.3.2.3.2 with the exception that "Timestamp" in "Measure" is undefined for this metric. If the reporting period is shorter than the "Initial_Buffering_Duration" then the client should send this parameter for each reporting period as long as it observes it. The "Value" field indicates the initial buffering duration where the unit of this metrics is expressed in seconds, and can be a fractional value. There can be only one "Measure" and it can only take one "Value". The absence of an event can be reported using the space (SP). "Initial_Buffering_Duration" is a session level parameter.

11.2.4 Successive loss of RTP packets

This parameter indicates the number of RTP packets lost in succession per media channel.

The syntax for the "Metrics-Name Successive_Loss" for the QoE-Feedback header is as defined in clause 5.3.2.3.2.

The absence of an event can be reported using the space (SP).

For the "Metrics-Name Successive_Loss", the "Value" field indicates the number of RTP packets lost in succession. The unit of this metric is expressed as an integer equal to or larger than 1. There is the possibility that successive loss occurs more than once during a reporting period. In that case the metrics value can occur more than once indicating the number of successive losses.

The optional "Timestamp" indicates the time when the succession of lost packets has occurred. The value of the "Timestamp" is equal to the NPT time of the last received RTP packet inside the reporting period, in playback order, before the occurrence of the succession of lost packets, relative to the starting time of the reporting period. If there is no received RTP packet inside the reporting period and before the succession of loss, the timestamp is set to the starting time of the reporting period.

If a full run length encoding of RTP losses with sequence number information is desired, RTCP XR [RFC 3611] Loss RLE Reporting Blocks should be used instead of the successive loss metric.

11.3 The QoE protocol

11.3.1 General

The RTSP and SDP based protocol extensions (see clauses 5.3.2.3 and 5.3.3.6) are used for transport and negotiation of the QoE metrics between the PSS client and the PSS server.

The QoE metrics negotiation starts with the response to the DESCRIBE request, if the metrics information is embedded in the SDP data (as described in example 1 in clause 11.3.2). For the case of locally stored SDP which contains QoE-Metrics attribute, the negotiation starts with client’s SETUP request. If the PSS client supports QoE metrics, then it shall send a SETUP request containing the selected (i.e. accepted by client)/modified (for re-negotiation) QoE metrics for either session level, or the media level, which is being set-up. Such a SETUP request is shown in example 2 in clause 11.3.3.

Upon receiving this SETUP request, the server shall return the RTSP Response with the "accepted" QoE metrics (i.e. metrics and metrics values which are identical to the ones in the client's request and accepted by the server) and the "re-negotiation" QoE metrics (i.e. metrics and metrics values which are not identical to the ones in the client's request and modified for re-negotiation by the server) .The echoing of the "accepted" QoE metrics is for re-acknowledging the client. The server may also reject the changes made by the client, i.e. reject the "re-negotiation" QoE metrics. If the server rejects the changes, it shall either set new values and resend the modified metrics back to the client, or it shall ignore the "re-negotiation" metrics and not re-acknowledge them. Any QoE metric that has been acknowledged as "accepted" by the server shall not be re-negotiated, i.e., it shall not be resent in the "3GPP-QoE-Metrics" header in the next RTSP request and shall not be re-acknowledged in the next RTSP response.

If the server does not approve the modifications done by the client, they should continue to re-negotiate until the RTSP PLAY request and the server shall echo the "accepted" QoE metrics in the RTSP PLAY response. A client can simply terminate the negotiation process by issuing an RTSP PLAY request. It must be noted that each time the "QoE-Metrics" header field is sent in an RTSP request, it shall also be present in the response corresponding to that particular request. Otherwise, the receiver of the response shall assume that the other end does NOT support QoE metrics.

If there is no DESCRIBE – RTSP Response pair sending at the beginning of the RTSP signalling (see Figure 11.2), it means that the SDP description is received by other means. If such an SDP contains the "3GPP-QoE-Metrics" attribute, the negotiation happens in the same way as it is described above, i.e. starts with SETUP request containing "3GPP-QoE-Metrics" header. If the SDP does not contain the "3GPP-QoE-Metrics" attribute and the server would still like to check whether the client supports QoE Protocol or not, the server shall include the "3GPP-QoE-Metrics" header containing the initial QoE metrics in the SETUP response. If the PSS client sends the QoE metrics information in the next request (indicating that it supports QoE Protocol), the negotiation shall continue until the mutual agreement is reached or RTSP PLAY request and response message pair is issued. If the client does not send QoE metrics information in the next request to SETUP response, then the server shall assume that the client does not support QoE metrics.

For performance and complexity reasons, QoE metrics renegotiation during streaming shall not be done. However it is possible to turn off the metrics during a streaming session. In clause 11.3 an example of messages, where the metrics are set to "Off" is given. The metrics can be set to "Off" at session level or at media level. The request url indicates what level is used. If no url is used, then "Off" applies to session level. The server should use OPTIONS (with Session ID) or SET_PARAMETER RTSP methods to turn off the QoE feedback.

A client should not send QoE feedback during RTSP ready state. After the ready state is ended (i.e., RTSP state=playing), the periodical feedback and normal operations continue. This reduces the network load in the uplink and downlink directions, and the processing overhead for the PSS client. When an RTSP PLAY request is sent by the PSS client after a PAUSE, the clock for measuring the reporting period (based on the defined "Sending Rate") shall be reset.

If there are multiple non-aggregated sessions, i.e. each media delivery is initiated by a different PLAY request, the QoE metrics are negotiated and reported for each session separately.

All the QoE Metrics in the following examples are fictitious. Clause 11.2 defines the actual QoE Metrics.

11.3.2 Metrics initiation with SDP

QoE metrics initiation with SDP shall be done according to clause 5.3.3.6.

This following example shows the syntax of the SDP attribute for QoE metrics. The session level QoE metrics description (Initial buffering duration and rebufferings) are to be monitored and reported only once at the end of the session. Also video specific description of metrics (corruptions and decoded bytes) are to be monitored and reported every 15 seconds from the beginning of the stream until the time 40s. Finally, audio specific description of metrics (corruptions) is to be monitored and reported every 20 seconds from the beginning until the end of the stream.

EXAMPLE 1:

```
S->C            RTSP/1.0 200 OK
                Cseq: 1
                Content-Type: application/sdp
                Content-Base: rtsp://example.com/foo/bar/baz.3gp/
                Content-Length: 800
                Server: PSSR6 Server

                v=0
                o=- 3268077682 433392265 IN IP4 63.108.142.6
                s=QoE Enables Session Description Example
                e=support@foo.com
                c=IN IP4 0.0.0.0
                t=0 0
                a=range:npt=0-83.660000
                a=3GPP-QoE-Metrics:{Initial_Buffering_Duration,Rebuffering_Duration};rate=End
                a=control:*
                m=video 0 RTP/AVP 96
                b=AS:28
                a=3GPP-QoE-Metrics:{Corruption_Duration,Decoded_Bytes};rate=15;range:npt=0-40
                a=control:trackID=3
                a=rtpmap:96 MP4V-ES/1000
                a=range:npt=0-83.666000
                a=fmtp:96profile-level-id=8;config=000001b008000001b50900012000
                m=audio 0 RTP/AVP 98
                b=AS:13
                a=3GPP-QoE-Metrics:{Corruption_Duration};rate=20
                a=control:trackID=5
                a=rtpmap:98 AMR/8000
                a=range:npt=0-83.660000
                a=fmtp:98 octet-align=1
                a=maxptime:200
```

11.3.3 Metrics initiation/termination with RTSP

QoE Metrics initiation with RTSP can be done according to clause 5.3.2.3.1

In the follwoing example it is shown how to negotiate QoE metrics during RTSP session setup.

EXAMPLE 2 (QoE metrics negotiation):

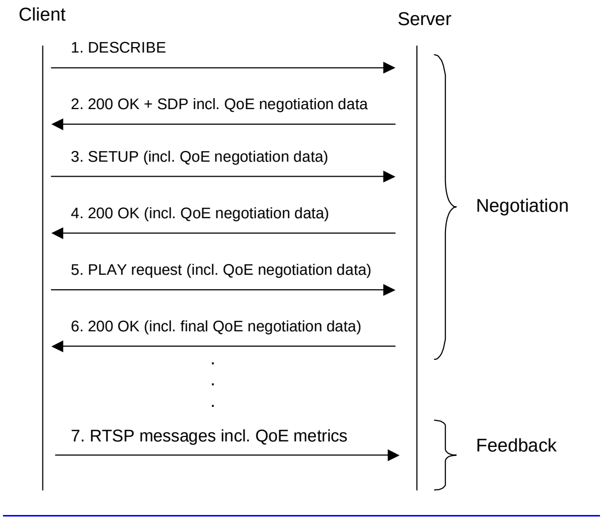

Client
Server
1. DESCRIBE
2. 200 OK + SDP incl. QoE negotiation data
3. SETUP (incl. QoE negotiation data)
4. 200 OK (incl. QoE negotiation data)
5. PLAY request (incl. QoE negotiation data)
6. 200 OK (incl. final QoE negotiation data)
7. RTSP messages incl. QoE metrics
Negotiation
Feedback

Figure 11.1: QoE metrics negotiation

```
C->S            SETUP rtsp://example.com/foo/bar/baz.3gp/trackID=3 RTSP/1.0
                Cseq: 2
                3GPP-QoE-Metrics:url=”rtsp://example.com/foo/bar/baz.3gp/trackID=3”;
                metrics={Corruption_Duration,Decoded_Bytes};rate=10; Range:npt=0-40,
                url=”rtsp://example.com/foo/bar/baz.3gp”;
                metrics={Initial_Buffering_Duration, Rebuffering_Duration};rate=End
```

In the above SETUP request, the client modifies the sending rate of the QoE metrics for the control URL ”rtsp://example.com/foo/bar/baz.3gp/trackID=3” from 15 to 10 (compared to the initial SDP description).

Assuming that the server acknowledged the changes, the server will send back a SETUP response as follows:

```
S->C            RTSP/1.0 200 OK
                Cseq: 2
                Session: 17903320
                Transport: RTP/AVP;unicast;client_port=7000-7001;server_port= 6970-6971
                3GPP-QoE-Metrics:url=”rtsp://example.com/foo/bar/baz.3gp/trackID=3”;
                metrics={Corruption_Duration,Decoded_Bytes};rate=10;Range:npt=0-40,
                url=”rtsp://example.com/foo/bar/baz.3gp”;
                metrics={Initial_Buffering_Duration,Rebuffering_Duration};rate=End
```

EXAMPLE 3 (QoE metrics negotiation – no DESCRIBE – 200/OK):

An example is shown in Figure 11.2 and can make use of the same RTSP header defined in clause 5.3.2.3.

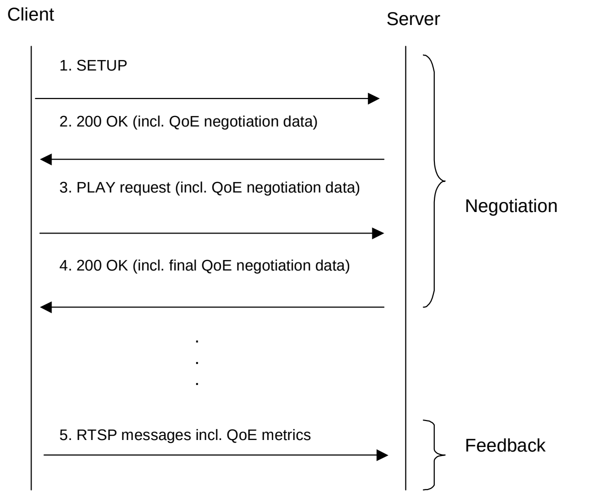

Client
Server
1. SETUP
2. 200 OK (incl. QoE negotiation data)
3. PLAY request (incl. QoE negotiation data)
4. 200 OK (incl. final QoE negotiation data)
Negotiation
5. RTSP messages incl. QoE metrics
Feedback

Figure 11.2: QoE metrics negotiation (no DESCRIBE-200/OK)

EXAMPLE 4 (setting the metrics off):

In this example, the metrics are switched off at session level (for all media).

```
C->S, S->C      SET_PARAMETER rtsp://example.com/foo/bar/baz.3gp RTSP/1.0
                Cseq: 302
                Session: 17903320
                3GPP-QoE-Metrics: Off
                Content-length: 0
```

The response for setting the metrics off would be:

```
S->C, C->S      RTSP/1.0 200 OK
                Cseq: 302
                Session: 17903320
                3GPP-QoE-Metrics: Off
```

11.3.4 Sending the metrics feedback with RTSP

QoE Metric feedback with RTSP can be formatted and sent according to clause 5.3.2.3.2.

The following example shows that during the monitoring time 2 corruption periods have occurred. Each value indicates the duration (in milliseconds) of each corruption period.

EXAMPLE 5 (Feedback):

```
C->S            SET_PARAMETER rtsp://example.com/foo/bar/baz.3gp RTSP/1.0
                Cseq: 302
                Session: 17903320
                3GPP-QoE-Feedback: 
                url=“rtsp://example.com/foo/bar/baz.3gp/trackID=3”;Corruption_Duration={200 1300} 
                Content-length: 0
```

The following example shows that during the monitoring time 2 corruption periods have occurred. Each values couple indicates the duration (in milliseconds) of each corruption period and the timestamp of the corruption (for example, the first corruption occurred at second 12 and lasted 200 milliseconds).

EXAMPLE 6 (Feedback with timestamps and range):

```
C->S            SET_PARAMETER rtsp://example.com/foo/bar/baz.3gp RTSP/1.0
                Cseq: 302
                Session: 17903320
                3GPP-QoE-Feedback: url=“rtsp://example.com/foo/bar/baz.3gp/trackID=3”; 
                Corruption_Duration={200 12, 1300 16};Range:npt=10-20
                Content-length: 0
```

In the following example there are no events to report.

EXAMPLE 7 (Feedback with no events):

```
C->S            SET_PARAMETER rtsp://example.com/foo/bar/baz.3gp RTSP/1.0
                Cseq: 302
                Session: 17903320
                3GPP-QoE-Feedback: 
                url=“rtsp://example.com/foo/bar/baz.3gp/trackID=3”;Corruption_Duration={ }
                Content-length: 0
```

Annex A (informative):
Protocols

A.1 SDP

This clause gives some background information on SDP for PSS clients.

Table A.1 provides an overview of the different SDP fields that can be identified in a SDP file. The order of SDP fields is mandated as specified in RFC 2327 [6].

Table A.1: Overview of fields in SDP for PSS clients

Type	Description		Requirement according to [6]	Requirement according to the present document
Session Description				
V	Protocol version		R	R
O	Owner/creator and session identifier		R	R
S	Session Name		R	R
I	Session information		O	O
U	URI of description		O	O
E	Email address		O	O
P	Phone number		O	O
C	Connection Information		R	R
B	Bandwidth information	AS	O	O
		RS	ND	O
		RR	ND	O
One or more Time Descriptions (See below)				
Z	Time zone adjustments		O	O
K	Encryption key		O	O
A	Session attributes	control	O	R
		range	O	R
		alt-group	ND	O
		3GPP-QoE-Metrics	ND	O
		3GPP-Asset-Information	ND	O
One or more Media Descriptions (See below)				
Time Description				
T	Time the session is active		R	R
R	Repeat times		O	O
Media Description				
M	Media name and transport address		R	R
I	Media title		O	O
C	Connection information		R	R
B	Bandwidth information	AS	O	R
		RS	ND	R
		RR	ND	R
K	Encryption Key		O	O
A	Attribute Lines	control	O	R
		range	O	R
		fmtp	O	R
		rtpmap	O	R
		X-predecbufsize	ND	O
		X-initpredecbufperiod	ND	O
		X-initpostdecbufperiod	ND	O
		X-decbyterate	ND	O
		framesize	ND	R (see note 5)
		alt	ND	O
		alt-default-id	ND	O
		3GPP-Adaptation-Support	ND	O
		3GPP-QoE-Metrics	ND	O
		3GPP-Asset-Information	ND	O

Note 1: R = Required, O = Optional, ND = Not Defined

Note 2: The "c" type is only required on the session level if not present on the media level.

Note 3: The "c" type is only required on the media level if not present on the session level.

Note 4: According to RFC 2327, either an 'e' or 'p' field must be present in the SDP description. On the other hand, both fields will be made optional in the future release of SDP. So, for the sake of robustness and maximum interoperability, either an 'e' or 'p' field shall be present during the server's SDP file creation, but the client should also be ready to receive SDP content containing neither 'e' nor 'p' fields.

Note 5: The "framesize" attribute is only required for H.263 streams.

Note 6: The “range” attribute is required on either session or media level: it is a session-level attribute unless the presentation contains media streams of different durations. If a client receives “range” on both levels, however, media level shall override session level.

The example below shows an SDP file that could be sent to a PSS client to initiate unicast streaming of a H.263 video sequence.

EXAMPLE 1:

```
v=0
o=ghost 2890844526 2890842807 IN IP4 192.168.10.10
s=3GPP Unicast SDP Example
i=Example of Unicast SDP file
u=http://www.infoserver.com/ae600
e=ghost@mailserver.com
c=IN IP4 0.0.0.0

t=0 0
a=range:npt=0-45.678
m=video 1024 RTP/AVP 96
b=AS:56~~128~~
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=control:rtsp://mediaserver.com/movie.3gp/trackID=1
a=framesize:96 176-144
a=recvonly
```

The following examples show some usage of the "alt" and the "alt-default-id" attributes (only the affected part of the SDP is shown):

EXAMPLE 2:

```
m=audio 0 RTP/AVP 97
b=AS:12
a=rtpmap:97 AMR/8000
a=control:trackID=1
a=fmtp:97 octet-align=1
a=range:npt=0-150.2
a=alt-default-id:1
a=alt:2:b=AS:16
a=alt:2:a=control:trackID=2
```

The equivalent SDP for alternative 1 (default) is:

EXAMPLE 3:

```
m=audio 0 RTP/AVP 97
b=AS:12
a=rtpmap:97 AMR/8000
a=control:trackID=1
a=fmtp:97 octet-align=1
a=range:npt=0-150.2
```

Alternative 2 is based on the default alternative but replaces two lines, "b=AS" and "a=control". Hence, the equivalent SDP for alternative 2 is:

```
EXAMPLE 4:    m=audio 0 RTP/AVP 97
              b=AS:16
              a=rtpmap:97 AMR/8000
              a=control:trackID=2
              a=fmtp:97 octet-align=1
              a=range:npt=0-150.2
```

Below is an example on the usage of the "alt-group" attribute with the subtype "BW":

```
EXAMPLE 5:    a=alt-group:BW:AS:32=1,4;56=2,4;64=3,5
```

The above line gives three groupings based on application-specific bitrate values. The first grouping will result in 32 kbps using media alternative 1 and 4. The second grouping has a total bitrate of 56 kbps using media alternatives 2 and 4. The last grouping needs 64 kbps when combing media alternatives 3 and 5.

Here follows an example on the usage of the "alt-group" attribute with the subtype "LANG":

```
EXAMPLE 6:    a=alt-group:LANG:RFC3066:en-US=1,2,4,5;se=3,4,5
```

The above line claims that media alternatives 1,2,4, and 5 supports US English and that media alternative 3, 4 and 5 supports Swedish.

A more complex example where a combination of "alt", "alt-default-id" and "alt-group" are used is seen below. The example allows a client to select a bandwidth that is suitable for the current context in an RTSP SETUP message. The client sends an RTSP DESCRIBE to the server and the server responds with the following SDP. A client, who supports the "alt", "alt-default-id" and "alt-group" attributes, can now select the most suitable alternative by using the control URLs corresponding to the selected alternatives in the RTSP SETUP message. The server sets up the selected alternatives and the client starts playing them. If the client is unaware of the attributes, they will be ignored. The result will be that the client uses the default "a=control" URLs at setup and receives the default alternatives.

```
EXAMPLE 7:    v=0
              o=ericsson_user 1 1 IN IP4 130.240.188.69
              s=A basic audio and video presentation
              c=IN IP4 0.0.0.0
              b=AS:56
              a=control:*
              a=range:npt=0-150.2
              a=alt-group:BW:AS:28=1,3;56=1,4;60=2,4;120=2,5
              t=0 0
              m=audio 0 RTP/AVP 97
              b=AS:12
              a=rtpmap:97 AMR/8000
              a=control:trackID=1
              a=fmtp:97 octet-align=1
              a=range:npt=0-150.2
              a=alt-default-id:1
              a=alt:2:b=AS:16
              a=alt:2:a=control:trackID=2
              m=video 0 RTP/AVP 98
              b=AS:44
              a=rtpmap:98 MP4V-ES/90000
              a=control:trackID=4
              a=fmtp:98 profile-level-id=8; config=01010000012000884006682C2090A21F
              a=range:npt=0-150.2
              a=X-initpredecbufperiod:98000
              a=alt-default-id:4
              a=alt:3:b=AS:16
              a=alt:3:a=control:trackID=3
              a=alt:3:a=X-initpredecbufperiod:48000
```

```
a=alt:5:b=AS:104
a=alt:5:a=control:trackID=5
a=alt:5:a=X-initpredecbufperiod:150000
```

The above example has 5 alternatives, 2 for audio and 3 for video. That would allow for a total of six combinations between audio and video. However, the grouping attribute recommends that only 4 of these combinations be used. The equivalent SDP for the default alternatives (alternatives 1 and 4) with a total session bitrate of 56 kbps follows:

```
EXAMPLE 8:    v=0
              o=ericsson_user 1 1 IN IP4 130.240.188.69
              s=Ericsson commercial
              c=IN IP4 0.0.0.0
              b=AS:56
              a=control:*
              a=range:npt=0-150.2
              t=0 0
              m=audio 0 RTP/AVP 97
              b=AS:12
              a=rtpmap:97 AMR/8000
              a=control:trackID=1
              a=fmtp:97 octet-align=1
              a=range:npt=0-150.2
              m=video 0 RTP/AVP 98
              b=AS:44
              a=rtpmap:98 MP4V-ES/90000
              a=control:trackID=4
              a=fmtp:98 profile-level-id=8; config=01010000012000884006682C2090A21F
              a=range:npt=0-150.2
              a=X-initpredecbufperiod:98000
```

The equivalent SDP for the 28 kbps total session bitrate (alternatives 1 and 3) is:

```
EXAMPLE 9:    v=0
              o=ericsson_user 1 1 IN IP4 130.240.188.69
              s=A basic audio and video presentation
              c=IN IP4 0.0.0.0
              b=AS:28
              a=control:*
              a=range:npt=0-150.2
              t=0 0
              m=audio 0 RTP/AVP 97
              b=AS:12
              a=rtpmap:97 AMR/8000
              a=control:trackID=1
              a=fmtp:97 octet-align=1
              a=range:npt=0-150.2
              m=video 0 RTP/AVP 98
              b=AS:16
              a=rtpmap:98 MP4V-ES/90000
              a=control:trackID=3
              a=fmtp:98 profile-level-id=8; config=01010000012000884006682C2090A21F
              a=range:npt=0-150.2
              a=X-initpredecbufperiod:48000
```

The equivalent SDP for the grouping with a 120 kbps total session bandwidth (alternatives 2 and 5):

```
EXAMPLE 10:   v=0
              o=ericsson_user 1 1 IN IP4 130.240.188.69
              s=A basic audio and video presentation
              c=IN IP4 0.0.0.0
              b=AS:120
```

```
a=control:*
a=range:npt=0-150.2
t=0 0
m=audio 0 RTP/AVP 97
b=AS:16
a=rtpmap:97 AMR/8000
a=control:trackID=2
a=fmtp:97 octet-align=1
a=range:npt=0-150.2
m=video 0 RTP/AVP 98
b=AS:104
a=rtpmap:98 MP4V-ES/90000
a=control:trackID=5
a=fmtp:98 profile-level-id=8; config=01010000012000884006682C2090A21F
a=range:npt=0-150.2
a=X-initpredecbufperiod:150000
```

The recommendation for a session with a total bitrate of 60 kbps is as easily formed. A client will use the received SDP and, as an example available bandwidth, to chose which alternatives to set up. If the client only has 32 kbps it selects the media alternatives 1 and 3, which use 28 kbps. The client sets this up by sending two normal RTSP requests using the control URLs from the chosen alternatives.

The audio SETUP request for the default (i.e. 56 kbps in the example above) looks like this:

```
EXAMPLE 11:   SETUP rtsp://media.example.com/examples/3G_systems.3gp/trackID=1 RTSP/1.0
              CSeq: 2
              Transport: RTP/AVP/UDP;unicast;client_port=3456-3457
```

The response from the server would be:

```
EXAMPLE 12:   RTSP/1.0 200 OK
              CSeq: 2
              Session: jEs.EdXCSKpB
              Transport: RTP/AVP/UDP;unicast;client_port=3456-3457;server_port=4002-4003;ssrc=5199dcb1
```

Also the video is added to the RTSP session under aggregated control:

```
EXAMPLE 13:   SETUP rtsp://media.example.com/examples/3G_systems.3gp/trackID=3 RTSP/1.0
              CSeq: 3
              Transport: RTP/AVP/UDP;unicast;client_port=3458-3459
              Session: jEs.EdXCSKpB
```

And the response would be:

```
EXAMPLE 14:   RTSP/1.0 200 OK
              CSeq: 3
              Session: jEs.EdXCSKpB
              Transport: RTP/AVP/UDP;unicast;client_port=3458-3459;server_port=4004-4005;ssrc=ae75904f
```

Had the client had more available bandwidth it could have set up another pair of alternatives in order to get better quality. The only change had been the RTSP URLs that had pointed at other media streams. For example the 120 kbps version would have been received if the audio SETUP request had used:

```
EXAMPLE 15:   rtsp://media.example.com/examples/3G_systems.3gp/trackID=2
```

and the video request

```
EXAMPLE 16:   rtsp://media.example.com/examples/3G_systems.3gp/trackID=5
```

The following example shows an SDP file that contains asset information, defined in Clause 5.3.3.7.

```
EXAMPLE 17:   v=0
              o=ghost 2890844526 2890842807 IN IP4 192.168.10.10
              s=3GPP Unicast SDP Example
              i=Example of Unicast SDP file
              u=http://www.infoserver.com/ae600
              e=ghost@mailserver.com
              c=IN IP4 0.0.0.0
              t=0 0
              a=range:npt=0-45.678
              a=3GPP-Asset-Information: {url="http://www.movie-database.com/title/thismovieinfo.xhtml"}
              a=3GPP-Asset-Information: {Title=MjhDRTA2NzI},{Copyright=Mjc0MkUwMUVGNDE2}
              m=video 1024 RTP/AVP 96
              b=AS:128
              a=rtpmap:96 H263-2000/90000
              a=fmtp:96 profile=3;level=10
              a=control:rtsp://mediaserver.com/movie.3gp/trackID=1
              a=framesize:96 176-144
              a=recvonly
```

A.2 RTSP

A.2.1 General

Clause 5.3.2 of the present document defines the required RTSP support in PSS clients and servers by making references to Appendix D of [5]. It also defines the RTSP header fields that are specific to PSS. The current clause gives an informative overview of these methods (see Table A.2) and headers (see Table A.3). Note that this overview does not replace the information in ~~that are specified in the referenced~~ Appendix D of [5] and Clause 5.3.2 of the present document, which must be consulted for a full implementation of RTSP in PSS. ~~An~~ Two examples of ~~an~~ RTSP sessions ~~is~~ are also given.

Table A.2: Overview of the ~~required~~ RTSP method support in PSS

Method	Requirement for a minimal on-demand playback client according to [5].	Requirement for a PSS client according to the present document.	Requirement for a minimal on-demand playback server according to [5].	Requirement for a PSS server according to the present document.
OPTIONS	O	O	Respond	Respond
REDIRECT	Respond	Respond	O	O
DESCRIBE	O	Generate	O	Respond
SETUP	Generate	Generate	Respond	Respond
PLAY	Generate	Generate	Respond	Respond
PAUSE	Generate	Generate	Respond	Respond
TEARDOWN	Generate	Generate	Respond	Respond
SET_PARAMETER	O	O	O	O
NOTE 1: O = Support is optional NOTE 2: 'Generate' means that the client/server is required to generate the request where applicable. NOTE 3: 'Respond' means that the client/server is required to properly respond to the request.				

Table A.3: Overview of the ~~required~~ RTSP header support in PSS

Header	Requirement for a minimal on-demand playback client according to [5].	Requirement for a PSS client according to the present document.	Requirement for a minimal on-demand playback server according to [5].	Requirement for a PSS server according to the present document.
Bandwidth	O	O	O	O
Connection	include/understand	include/understand	include/understand	include/understand
Content-Encoding	understand	understand	include	include
Content-Language	understand	understand	include	include
Content-Length	understand	understand	include	include
Content-Type	understand	understand	include	include
CSeq	include/understand	include/understand	include/understand	include/understand
Date	include	include	include	include
Location	understand	understand	O	O
Public	O	O	include	include
Range	O	include/understand	understand	include/understand
Require	O	O	understand	understand
RTP-Info	understand	understand	include	include
Server[4]	O	O	O	O
Session	include	include	understand	understand
Timestamp	O	O	include/understand	include/understand
Transport	include/understand	include/understand	include/understand	include/understand
Unsupported	include	include	include	include
User-Agent[4]	O	O	O	O
3GPP-Adaptation	N/A	O	N/A	O
3GPP-Link-Char	N/A	O	N/A	O
3GPP-QoE-Metrics	N/A	O	N/A	O

NOTE 1: O = Support is optional

NOTE 2: 'include' means that the client/server is required to include the header in a request or response where applicable.

NOTE 3: 'understand' means that the client/server is required to be able to respond properly if the header is received in a request or response.

NOTE 4: According to [5] the "Server" and “User-Agent” headers ~~is~~ are not strictly required for a minimal RTSP ~~client~~ implementation, although it is highly recommended that ~~it is~~ they are included with responses and requests. The same applies to ~~a~~ PSS servers and clients according to the present document.

The example below is intended to give some more understanding of how RTSP and SDP are used within the 3GPP PSS. The example assumes that the streaming client has the RTSP URL to a presentation consisting of an H.263 video sequence and AMR speech. RTSP messages sent from the client to the server are in **bold** and messages from the server to the client in *italic*. In the example the server provides aggregate control of the two streams.

EXAMPLE 1:

DESCRIBE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 1
User-Agent: TheStreamClient/1.1b2

RTSP/1.0 200 OK
CSeq: 1
Content-Type: application/sdp
Content-Length: 435

v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video
e=foo@bar.com

c=IN IP4 0.0.0.0
b=AS:77
t=0 0
a=range:npt=0-59.3478
a=control:*
m=audio 0 RTP/AVP 97

```
b=AS:13
b=RR:350
b=RS:300
a=rtpmap:97 AMR/8000
a=fmtp:97
a=maxptime:200
a=control:streamID=0
m=video 0 RTP/AVP 98
b=AS:64
b=RR:2000
b=RS:1200
a=rtpmap:98 H263-2000/90000
a=fmtp:98 profile=3;level=10
a=control: streamID=1
```

```
SETUP rtsp://mediaserver.com/movie.test/streamID=0 RTSP/1.0
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457
User-Agent: TheStreamClient/1.1b2
```

```
RTSP/1.0 200 OK
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457; server_port=5678-5679
Session: dfhyrio90llk
```

```
SETUP rtsp://mediaserver.com/movie.test/streamID=1 RTSP/1.0
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
```

```
RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459; server_port=5680-5681
Session: dfhyrio90llk
```

```
PLAY rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 4
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
```

```
RTSP/1.0 200 OK
CSeq: 4
Session: dfhyrio90llk
Range: npt=0-
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0; seq=9900;rtptime=4470048,
          url= rtsp://mediaserver.com/movie.test/streamID=1; seq=1004;rtptime=1070549
```

NOTE: Headers can be folded onto multiple lines if the continuation line begins with a space or horizontal tab. For more information, see RFC2616 [17].

The user watches the movie for 20 seconds and then decides to fast forward to 10 seconds before the end…

```
PAUSE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 5
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
```

```
PLAY rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 6
Range: npt=50-59.3478
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
```

```
RTSP/1.0 200 OK
CSeq: 5
Session: dfhyrio90llk
```

```
RTSP/1.0 200 OK
CSeq: 6
Session: dfhyrio90llk
Range: npt=50-59.3478
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0;
              seq=39900;rtptime=44470648,
               url= rtsp://mediaserver.com/movie.test/streamID=1;
              seq=31004;rtptime=41090349
```

After the movie is over the client issues a TEARDOWN to end the session…

```
TEARDOWN rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 7
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
```

```
RTSP/1.0 200 OK
Cseq: 7
Session: dfhyrio90llk
Connection: close
```

The example below contains a complete RTSP signalling for session set-up with rate adaptation support, where the client buffer feedback functionality is initialised and used. To allow the server to know that a client supports the buffer feedback formats and signalling, the client includes a link to its UAProf description in its RTSP DESCRIBE request.

EXAMPLE 2:

```
DESCRIBE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 1
User-Agent: TheStreamClient/1.1b2
x-wap-profile: http://uaprof.example.com/products/TheStreamClient1.1b2
```

```
RTSP/1.0 200 OK
CSeq: 1Date: 20 Aug 2003 15:35:06 GMT
Content-Base: rtsp://mediaserver.com/movie.test/
Content-Type: application/sdp
Content-Length: 500

v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video
e=foo@bar.com
c=IN IP4 0.0.0.0
b=AS:77
t=0 0
a=range:npt=0-59.3478
a=control:*
m=audio 0 RTP/AVP 97
b=AS:13
b=RR:350
b=RS:300
a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align=1
a=control: streamID=0
a=3GPP-Adaptation-Support:2
m=video 0 RTP/AVP 98
b=AS:64
b=RR:2000
b=RS:1200
a=rtpmap:98 H263-2000/90000
a=fmtp:98 profile=3;level=10
a=control: streamID=1
a=3GPP-Adaptation-Support:1


SETUP rtsp://mediaserver.com/movie.test/streamID=0 RTSP/1.0
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457
User-Agent: TheStreamClient/1.1b2
3GPP-Adaptation: url="rtsp://mediaserver.com/movie.test/streamID=0";size=14500;target-time=5000


RTSP/1.0 200 OK
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457;server_port=5678-5679;ssrc=A432F9B1
Session: dfhyrio90llk
3GPP-Adaptation: url="rtsp://mediaserver.com/movie.test/streamID=0";size=14500;target-time=5000


SETUP rtsp://mediaserver.com/movie.test/streamID=1 RTSP/1.0
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
3GPP-Adaptation: url="rtsp://mediaserver.com/movie.test/streamID=1";size=35000;target-time=5000
```

```
RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459; server_port=5680-5681; ssrc=4D23AE29
Session: dfhyrio90llk
3GPP-Adaptation: url=" rtsp://mediaserver.com/movie.test/streamID=1";size=35000;target-time=5000
```

```
PLAY rtsp://mediaserver.com/movie.test/ RTSP/1.0
CSeq: 4
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
```

```
RTSP/1.0 200 OK
CSeq: 4
Session: dfhyrio90llk
Range: npt=0-
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0; seq=9900;rtptime=4470048, url= rtsp://mediaserver.com/movie.test/streamID=1; seq=1004;rtptime=1070549
```

If the client desires to change the target buffer protection time during the session, it can signal a new value to the server by means of an RTSP SET_PARAMETER request.

```
SET_PARAMETER rtsp://mediaserver.com/movie.test/ RTSP/1.0
CSeq: 8
Session: dfhyrio90llk
User-Agent: TheStreamClient/1.1b2
3GPP-Adaptation: url="rtsp://mediaserver.com/movie.test/streamID=0";target-time=7000,url="rtsp://mediaserver.com/movie.test/streamID=1";target-time=7000
```

```
RTSP/1.0 200 OK
CSeq: 8
Session: dfhyrio90llk
3GPP-Adaptation: url="rtsp://mediaserver.com/movie.test/streamID=0";target-time=7000,url="rtsp://mediaserver.com/movie.test/streamID=1";target-time=7000
```

A.2.2 Implementation guidelines

A.2.2.1 Usage of persistent TCP

Considering the potentially long round-trip-delays in a packet switched streaming service over UMTS it is important to keep the number of messages exchanged between a server and a client low. The number of requests and responses exchanged is one of the factors that will determine how long it takes from the time that a user initiates PSS until the streams starts playing in a client.

RTSP methods are sent over either TCP or UDP for IP. Both client and server shall support RTSP over TCP whereas RTSP over UDP is optional. For TCP the connection can be persistent or non-persistent. A persistent connection is used for several RTSP request/response pairs whereas one connection is used per RTSP request/response pair for the non-persistent connection. In the non-persistent case each connection will start with the three-way handshake (SYN, ACK, SYN) before the RTSP request can be sent. This will increase the time for the message to be sent by one round trip delay.

For these reasons it is recommended that 3GPP PSS clients should use a persistent TCP connection, at least for the initial RTSP methods until media starts streaming.

A.2.2.2 Detecting link aliveness

In the wireless environment, connection may be lost due to fading, shadowing, loss of battery power, or turning off the terminal even though the PSS session is active. In order for the server to be able to detect the client's aliveness, the PSS client should send "wellness" information to the PSS server for a defined interval as described in the RFC2326. There are several ways for detecting link aliveness described in the RFC2326, however, the client should be careful about issuing "PLAY method without Range header field" too close to the end of the streams, because it may conflict with pipelined PLAY requests. Below is the list of recommended "wellness" information for the PSS clients and servers in a prioritised order.

1. RTCP
2. OPTIONS method with Session header field

NOTE: Both servers and clients can initiate this OPTIONS method.

The client should send the same wellness information in 'Ready' state as in 'Playing' and 'Recording' states, and the server should detect the same client's wellness information in 'Ready' state as in 'Playing' and 'Recording' states. In particular, the same link aliveness mechanism should be managed following a 'PAUSE' request and response.

A.3 RTP

A.3.1 General

Void.

A.3.2 Implementation guidelines

A.3.2.1 Maximum RTP packet size

The RFC 3550 (RTP) [9] does not impose a maximum size on RTP packets. However, when RTP packets are sent over the radio link of a 3GPP PSS system there is an advantage in limiting the maximum size of RTP packets.

Two types of bearers can be envisioned for streaming using either acknowledged mode (AM) or unacknowledged mode (UM) RLC. The AM uses retransmissions over the radio link whereas the UM does not. In UM mode large RTP packets are more susceptible to losses over the radio link compared to small RTP packets since the loss of a segment may result in the loss of the whole packet. On the other hand in AM mode large RTP packets will result in larger delay jitter compared to small packets as there is a larger chance that more segments have to be retransmitted.

For these reasons it is recommended that the maximum size of RTP packets should be limited in size taking into account the wireless link. This will decrease the RTP packet loss rate particularly for RLC in UM. For RLC in AM the delay jitter will be reduced permitting the client to use a smaller receiving buffer. It should also be noted that too small RTP packets could result in too much overhead if IP/UDP/RTP header compression is not applied or unnecessary load at the streaming server.

In the case of transporting video in the payload of RTP packets it may be that a video frame is split into more than one RTP packet in order not to produce too large RTP packets. Then, to be able to decode packets following a lost packet in the same video frame, it is recommended that synchronisation information be inserted at the start of such RTP packets. For H.263 this implies the use of GOBs with non-empty GOB headers and in the case of MPEG-4 video the use of video packets (resynchronisation markers). If the optional Slice Structured mode (Annex K) of H.263 is in use, GOBs are replaced by slices.

A.3.2.2 Sequence number and timestamp in the presence of NPT jump

The description below is intended to give more understanding of how RTP sequence number and timestamp are specified within the 3GPP PSS in the presence of NPT jumps. The jump happens when a client sends a PLAY request to skip media.

The RFC 2326 (RTSP) [5] specifies that both RTP sequence numbers and RTP timestamps must be continuous and monotonic across jumps of NPT. Thus when a server receives a request for a skip of the media that causes a jump of NPT, it shall specify RTP sequence numbers and RTP timestamps continuously and monotonically across the skip of the media to conform to the RTSP specification. Also, the server may respond with "seq" in the RTP-Info field if this parameter is known at the time of issuing the response.

A.3.2.3 RTCP transmission interval

In RTP [9] when using the basic RTP profile AVP [10], Section 6.2~~,~~ of [9] defines rules for the calculation of the interval between the sending of two consecutive RTCP packets, i.e. the RTCP transmission interval~~, are defined~~. These rules consist of two steps:

- Step 1: an algorithm that calculates a transmission interval from parameters such as the RTCP bandwidth defined in section 5.3.3.1 and the average RTCP packet size. This algorithm is described in [9], with example code in annex A.7.

- Step 2: Taking the maximum of the transmission interval computed in step 1 and a mandatory fixed minimum RTCP transmission interval~~ of 5 seconds~~. The RTP/RTCP specification [9] gives a recommendation that the minimum interval is set to 5 seconds, but it may be scaled to other values in unicast sessions for all participants (SSRCs), see section 6.2 of [9] for further details. For PSS and the AVP profile the minimum interval shall be 5 seconds.

NOTE: The algorithm in Annex A.7 of [9] must be accordingly modified to enable usage of the explicit bandwidth values given for the RTCP bandwidth, as provided by the SDP bandwidth modifiers (RR and RS) that shall be used by PSS according to clause 5.3.3.1.

Implementations conforming to this TS shall perform step 1 and may perform step 2. All other algorithms and rules of [9] stay valid and shall be followed. Please note that the processing described in [9] include a randomisation with an equally distributed random function resulting in a value somewhere between 0.5 to 1.5 times the calculated value prior to further scaling with a factor of 1/(e-1.5). Those RTCP intervals either can be compared as the average value or as the maximum interval.

The rules defined in RTP [9] and AVP [10] are updated by the AVPF profile [57]. The new rules remove the minimum transmission interval rule. It also provides SDP signalling that allows the server to configure the RTCP behaviour. When using the AVPF profile the PSS client and server shall send RTCP according to the rules in [57] and comply with the signalled parameters.

~~Following these recommendations results in regular sending of RTCP messages, where the interval between those is depending on the RTCP bandwidth and the average RTCP packet size.~~Below are formulas for calculating the maximal RTCP interval for given input parameters. Normally the RTCP packets will be sent with smaller intervals. The formulas below have been reduced as much as possible and utilize the rules resulting in the largest interval. The formulas are not a replacement for implementing the algorithm in any stack, as some of the input values are dynamic and will change during a session.

Variables:

RSv:	The RTCP bandwidth in bits/s assigned to active data senders
RRv:	The RTCP bandwidth in bits/s assigned to data receiver only.
members:	The total number of participants (SSRCs) in the session.
avg_rtcp_size:	The average RTCP packet size in bytes.
min_rtcp_interval:	The minimum RTCP transmission interval in seconds.
t_rr_interval:	The minimum reporting interval in seconds when in regular RTCP mode for AVPF.

The calculation for the AVP profile:

x = 1.5 * max((avg_rtcp_size * 8 * members / min(RSv, RRv)), min_rtcp_interval) / 1.21828

The calculation for the AVPF profile:

x =1.5 * max(2*(avg_rtcp_size * 8 * members / min(RSv, RRv)) / 1.21828, t_rr_interval)

The above formulas are valid for both a PSS server and a PSS client, and either side can compute the maximum RTCP interval of either of the two sides. For example, the PSS server can compute the maximum RTCP transmission interval for the RTCP packets received by the PSS client just by replacing the expression min(RSv, RRv) with RRv in the formula.

When using the AVPF profile the sending of RTCP reports is governed by the AVPF mode in use, the RTCP bandwidth, the average RTCP packet size and possibly the minimal reporting interval (t_rr_interval). In AVPF the RTCP sender will work in regular reporting mode, unless there are any events to report on. This means that the normal bandwidth limitation rule is used, possibly combined with suppression based on the t_rr_interval variable. The t_rr_interval variable can be set using signalling in SDP with the "trr-int" parameter. Also, due to the transitions between early RTCP mode and the regular reporting mode the reporting can be delayed a complete regular reporting interval. The other modes will all send RTCP at least as often as for the transition between early and regular mode.

A.3.2.4 Timestamp handling after PAUSE/PLAY requests

The description below intends to clarify how RTP timestamps are specified within the 3GPP PSS when a client sends a PLAY request following a PAUSE request. The RTP timestamp space must be continuous along time during a session and then reflect the actual time elapsed since the beginning of the session. A server must reflect the actual time interval elapsed between the last RTP packets sent before the reception of the PAUSE request and the first RTP packets sent after the reception of the PLAY request in the RTP timestamp. A client will need to compute the mapping between NPT time and RTP timestamp each time it receives a PLAY response for on-demand content. This means that a client must be able to cope with any gap in RTP timestamps after a PLAY request.

The PLAY request can include a Range header if the client wants to seek backward or forward in the media, or without a Range header if the client only wants to resume the paused session.

Example:
In this example Client C plays a media file from Server S. RTP timestamp rate in this example is 1000Hz for clarity.

```
C -> S:   PLAY rtsp://example.com/mediastream RTSP/1.0
    CSeq: 2
    Session: 123456
    Range: npt=1.125-


S -> C:   RTSP/1.0 200 OK
    CSeq: 2
    Session: 123456
    Range: npt=1.120-
    RTP-Info: url=rtsp://example.com/mediastream;seq=1000;rtptime=5000


S -> C:   RTP packet - seq = 1000 - rtptime = 5000 - corresponding media time (NPT time) =  1120ms
S -> C:   RTP packet - seq = 1001 - rtptime = 5040 - corresponding media time (NPT time) =  1160ms
S -> C:   RTP packet - seq = 1002 - rtptime = 5080 - corresponding media time (NPT time) =  1200ms
S -> C:   RTP packet - seq = 1003 - rtptime = 5120 - corresponding media time (NPT time) =  1240ms


C -> S: PAUSE rtsp://example.com/mediastream RTSP/1.0
    CSeq: 3
    Session: 123456


S -> C:   RTSP/1.0 200 OK
    CSeq: 3
    Session: 123456


[10 seconds elapsed]


C -> S:   PLAY rtsp://example.com/mediastream RTSP/1.0
    CSeq: 4
```

Session: 123456

S -> C: RTSP/1.0 200 OK
CSeq: 4
Session: 123456
Range: npt=1.280-
RTP-Info: url=rtsp://example.com/mediastream;seq=1004;rtptime=15160

S -> C: RTP packet - seq = 1004 - rtptime = 15160 - corresponding media time (NPT time) = 1280ms
S -> C: RTP packet - seq = 1005 - rtptime = 15200 - corresponding media time (NPT time) = 1320ms
S -> C: RTP packet - seq = 1006 - rtptime = 15240 - corresponding media time (NPT time) = 1360ms

C -> S: PAUSE rtsp://example.com/mediastream RTSP/1.0
CSeq: 5
Session: 123456

S -> C: RTSP/1.0 200 OK
CSeq: 5
Session: 123456

C -> S: PLAY rtsp://example.com/mediastream RTSP/1.0
CSeq: 6
Session: 123456
Range: npt=0.5-

[55 milliseconds elapsed during request processing]

S -> C: RTSP/1.0 200 OK
CSeq: 6
Session: 123456
Range: npt=0.480-
RTP-Info: url=rtsp://example.com/mediastream;seq=1007;rtptime=15295

S -> C: RTP packet - seq = 1007 - rtptime = 15295 - corresponding media time (NPT time) = 480ms
S -> C: RTP packet - seq = 1008 - rtptime = 15335 - corresponding media time (NPT time) = 520ms
S -> C: RTP packet - seq = 1009 - rtptime = 15375 - corresponding media time (NPT time) = 560ms

A.3.3 Examples of RTCP APP packets for client buffer feedback

Example 1: The RTCP Receiver Report and OBSN packet while having a number of packets for a single source in the receiver buffer and signalling the playout delay for the oldest packet.

RTCP Receiver Report:

```
 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|    RC   |   PT=RR=201   |             length = 7        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SSRC of packet sender = 0x324FE239        |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_1 (SSRC of first source) = 0x4D23AE29    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| fraction lost |       cumulative number of packets lost       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| extended highest sequence number received = 0x00000551 (1361) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

APP packet:

```
 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|subtype=0|   PT=APP=204  |             length = 4        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Client SSRC = 0x324FE239                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          name = "PSS0"                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Server SSRC = 0x4D23AE29                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Playout Delay = 300     |            OBSN = 1323        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

From the above compound RTCP packet, the server concludes that the client has 39 (1361-1323+1) packets in its video buffer, which has a total size of 35000 bytes as indicated during the RTSP session setup (see rate-adaptation example in clause A.2.1).

The server can compute the buffer duration at the time the packet was sent by first computing the time difference between the timestamp of the packet of highest sequence number (i.e. sequence number 1361) and the timestamp of the packet of oldest sequence number (i.e. sequence number 1323) and second adding the playout delay of the oldest packet (300).

If the receiver had chosen not to signal the playout delay of the oldest packet, the receiver would have sent instead the reserved value 0x FFFF for the playout delay field.

Example 2: Reporting an empty buffer.

In the case a client has played out all packets for a SSRC that has been received and would send out a RTCP receiver report according to the one in example 1, the OBSN packet would carry an OBSN value of 1362. This results in that the calculation of the number of packets becomes 0 (1361-1362+1). As the buffer is empty, the playout delay is not defined and the receiver should use the reserved value 0xFFFF for this field.

A.4 Capability exchange

A.4.1 Overview

Clause A.4 provides detailed information about the structure and exchange of device capability descriptions for the PSS. It complements the normative part contained in clause 5.2 of the present document.

The functionality is sometimes referred to as capability exchange. Capability exchange in PSS uses the CC/PP [39] framework and reuse parts of the CC/PP application UAProf [40].

To facilitate server-side content negotiation for streaming, the PSS server needs to have access to a description of the specific capabilities of the mobile terminal, i.e. the device capability description. The device capability description contains a number of attributes. During the set-up of a streaming session the PSS server can use the description to provide the mobile terminal with the correct type of multimedia content. Concretely, it is envisaged that servers use information about the capabilities of the mobile terminal to decide which stream(s) to provision to the connecting terminal. For instance, the server could compare the requirements on the mobile terminal for multiple available variants of a stream with the actual capabilities of the connecting terminal to determine the best-suited stream(s) for that particular terminal. A similar mechanism could also be used for other types of content.

A device capability description contains a number of device capability attributes. In the present document they are referred to as just attributes. The current version of PSS does not include a definition of any specific user preference attributes. Therefore we use the term device capability description. However, it should be noted that even though no specific user preference attributes are included, simple tailoring to the preferences of the user could be achieved by temporarily overrides of the available attributes. E.g. if the user for a particular session only would like to receive mono sound even though the terminal is capable of stereo, this can be accomplished by providing an override for the "AudioChannels" attribute. It should also be noted that the extension mechanism defined would enable an easy introduction of specific user preference attributes in the device capability description if needed.

The term device capability profile or profile is sometimes used instead of device capability description to describe a description of device capabilities and/or user preferences. The three terms are used interchangeably in the present document.

Figure A.1 illustrates how capability exchange in PSS is performed. In the simplest case the mobile terminal informs the PSS server(s) about its identity so that the latter can retrieve the correct device capability profile(s) from the device profile server(s). For this purpose, the mobile terminal adds one or several URLs to RTSP and/or HTTP protocol data units that it sends to the PSS server(s). These URLs point to locations on one or several device profile servers from where the PSS server should retrieve the device capability profiles. This list of URLs is encapsulated in RTSP and HTTP protocol data units using additional header field(s). The list of URLs is denoted URLdesc. The mobile terminal may supplement the URLdesc with extra attributes or overrides for attributes already defined in the profile(s) located at URLdesc. This information is denoted Profdiff. As URLdesc, Profdiff is encapsulated in RTSP and HTTP protocol data units using additional header field(s).

The device profile server in Figure A.1 is the logical entity that stores the device capability profiles. The profile needed for a certain request from a mobile terminal may be stored on one or several such servers. A terminal manufacturer or a software vendor could maintain a device profile server to provide device capability profiles for its products. It would also be possible for an operator to manage a device profile server for its subscribers and then e.g. enable the subscriber to make user specific updates to the profiles. The device profile server provides device capability profiles to the PSS server on request.

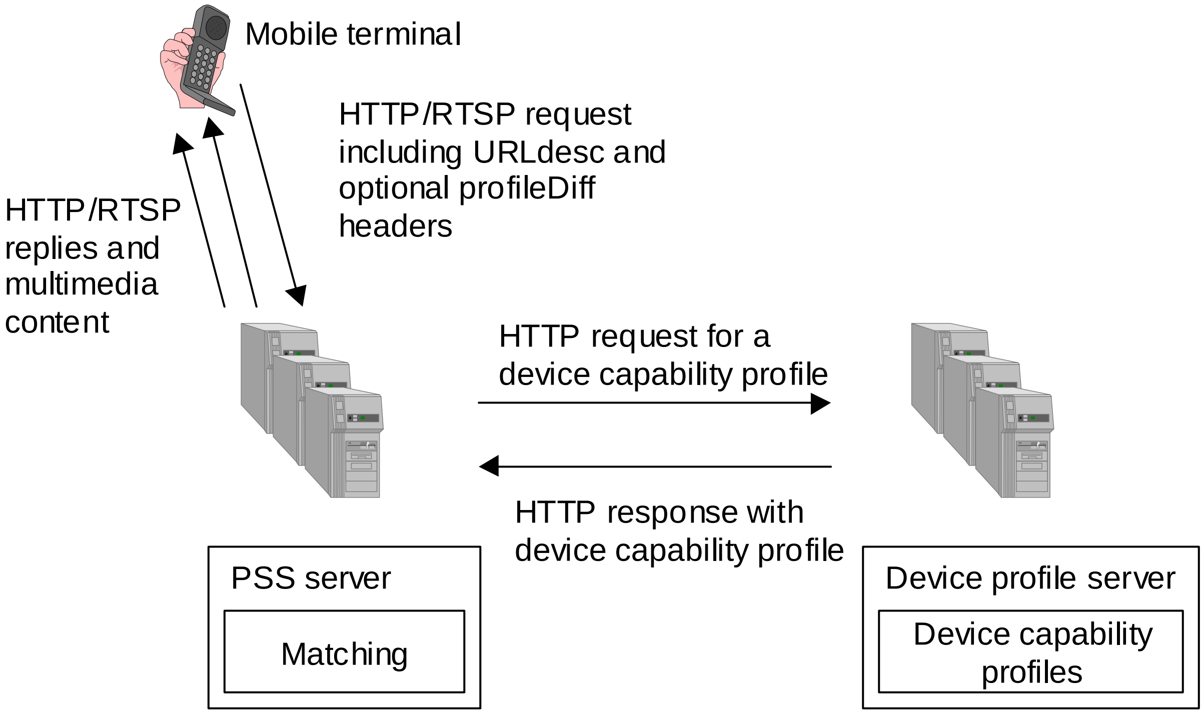

Mobile terminal
HTTP/RTSP request including URLdesc and optional profileDiff headers
HTTP/RTSP replies and multimedia content
HTTP request for a device capability profile
HTTP response with device capability profile
PSS server
Matching
Device profile server
Device capability profiles

Figure A.1: Functional components in PSS capability exchange

The PSS server is the logical entity that provides multimedia streams and other, static content (e.g. SMIL documents, images, and graphics) to the mobile terminal (see Figure A.1). A PSS application might involve multiple PSS servers, e.g. separate servers for multimedia streams and for static content. A PSS server handles the matching process.

Matching is a process that takes place in the PSS servers (see Figure A.1). The device capability profile is compared with the content descriptions at the server and the best fit is delivered to the client.

A.4.2 Scope of the specification

The following bullet list describes what is considered to be within the scope of the specification for capability exchange in PSS.

- Definition of the structure for the device capability profiles, see clause A.4.3.
- Definition of the CC/PP vocabularies, see clause A.4.4.
  - Reference to a set of device capability attributes for multimedia content retrieval applications that have already been defined by UAProf [40]. The purpose of this reference is to point out which attributes are useful for the PSS application.
  - Definition of a set of device capability attributes specifically for PSS applications that are missing in UAProf.
- It is important to define an extension mechanism to easily add attributes since it is not possible to cover all attributes from the beginning. The extension mechanism is described in clause A.4.5.
- The structure of URLdesc, Profdiff and their interchange is described in clause A.4.6.
- Protocols for the interchange of device capability profiles between the PSS server and the device profile server is defined in clause 5.2.7.

The specification does not include:

- rules for the matching process on the PSS server. These mechanisms should be left to the implementations. For interoperability, only the format of the device capability description and its interchange is relevant.
- definition of specific user preference attributes. It is very difficult to standardise such attributes since they are dependent on the type of personalised services one would like to offer the user. The extensible descriptions format and exchange mechanism proposed in this document provide the means to create and exchange such attributes if needed in the future. However, as explained in clause A.4.1 limited tailoring to the preferences of the user could be achieved by temporarily overriding available attributes in the vocabularies already defined for PSS. The vocabulary also includes some very basic user preference attributes. For example, the profile includes a list of preferred languages. Also the list of MIME types can be interpreted as user preference, e.g. leaving out audio MIME's could mean that user does not want to receive any audio content. The available attributes are described in clause 5.2.3 of the present document.
- requirements for caching of device capability profiles on the PSS server. In UAProf, a content server can cache the current device capability profile for a given WSP session. This feature relies on the presence of WSP sessions. Caching significantly increases the complexity of both the implementations of the mobile terminal and the server. However, HTTP is used between the PSS server and the device profile server. For this exchange, normal content caching provisions as defined by HTTP apply and the PSS server may utilise this to speed up the session set-up (see clause 5.2.7)
- intermediate proxies. This feature is considered not relevant in the context of PSS applications.

A.4.3 The device capability profile structure

A device capability profile is a description of the capabilities of the device and possibly also the preferences of the user of that device. It can be used to guide the adaptation of content presented to the device. A device capability profile for PSS is an RDF [41] document that follows the structure of the CC/PP framework [39] and the CC/PP application UAProf [40]. The terminology of CC/PP is used in this text and therefore briefly described here.

-Attributes are used for specifying the device capabilities and user preferences. A set of attribute names, permissible values and semantics constitute a CC/PP vocabulary. An RDF schema defines a vocabulary. The syntax of the attributes is defined in the schema but also, to some extent, the semantics. A profile is an instance of a schema and contains one or more attributes from the vocabulary. Attributes in a schema are divided into components distinguished by attribute characteristics. In the CC/PP specification it is anticipated that different applications will use different vocabularies. According to the CC/PP framework a hypothetical profile might look like Figure A.2. A further illustration of how a profile might look like is given in the example in clause A.4.7.

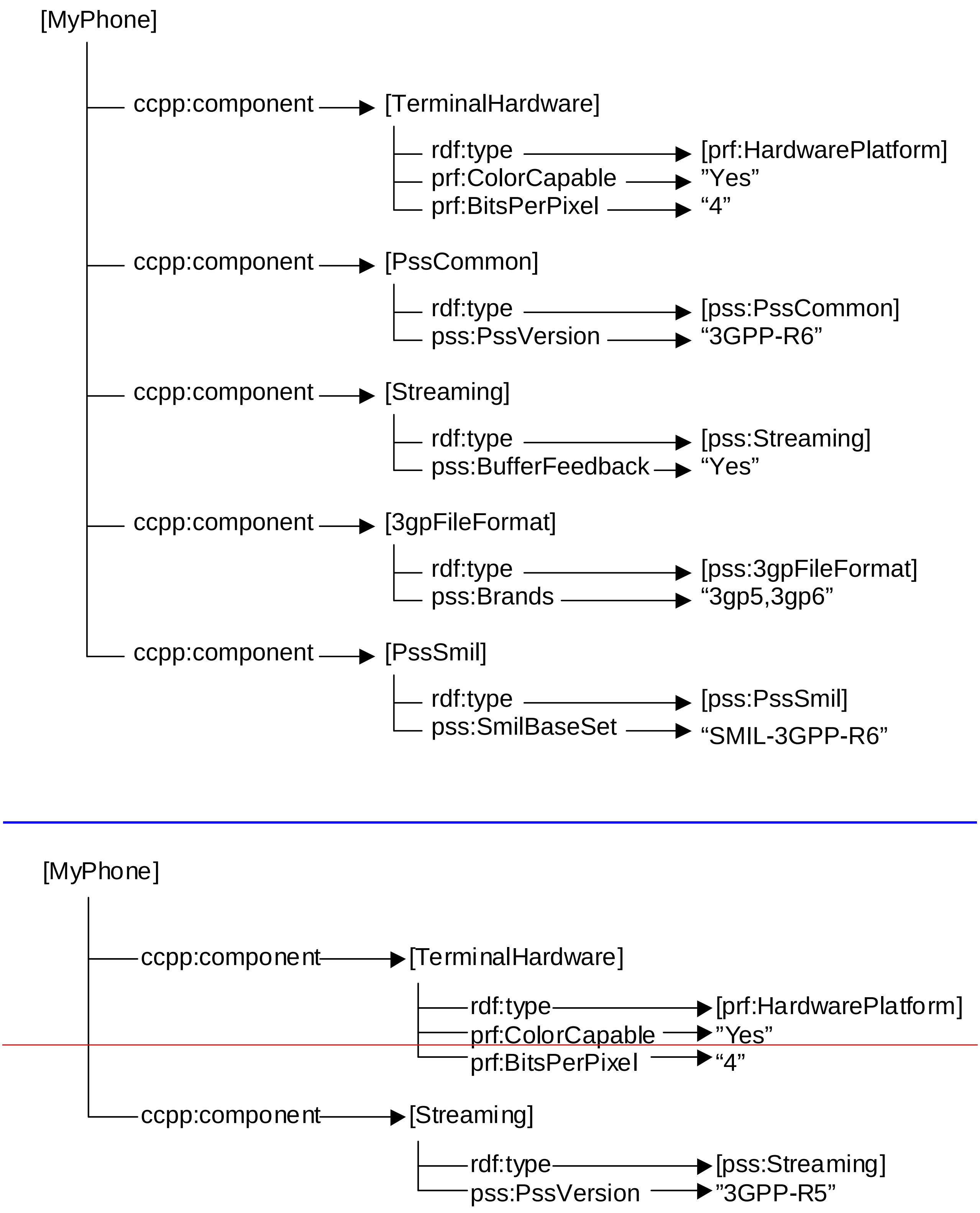


Figure A.2: Illustration of the profile structure

A CC/PP schema is extended through the introduction of new attribute vocabularies and a device capability profile can use attributes drawn from an arbitrary number of different vocabularies. Each vocabulary is associated with a unique XML namespace. This mechanism makes it possible to reuse attributes from other vocabularies. It should be mentioned that the prefix **ccpp** identifies elements of the CCPP namespace (URI http://www.w3.org/2002/11/08-ccpp-ns~~1999/02/22-rdf-syntax-ns~~), **prf** identifies elements of the UAProf namespace (URI http://www.wapforum.org/profiles/UAPROF/ccppschema-20010330) ,~~-~~**rdf** identifies elements of the RDF namespace (URI http://www.w3.org/1999/02/22-rdf-syntax-ns) and **pss** identifies elements of the PSS Release-6 ~~Streaming~~ namespace. (URI http://www.3gpp.org/profiles/PSS/ccppschema-PSS6~~5~~).

Attributes of a component can be included directly or may be specified by a reference to a CC/PP default profile. Resolving a profile that includes a reference to a default profile is time-consuming. When the PSS server receives the profile from a device profile server the final attribute values can not be determined until the default profile has been requested and received. Support for defaults is required by the CC/PP specification [39]. Due to these problems, there is a recommendation made in clause 5.2.6 to not use the CC/PP defaults element in PSS device capability profile documents.

A.4.4 CC/PP Vocabularies

A CC/PP vocabulary shall according to CC/PP and UAProf include:

- a~~A~~n RDF schema for the vocabulary based on the CC/PP schema;~~.~~
- a~~A~~ description of the semantics/type/resolution rules/sample values for each attribute;~~.~~
- a~~A~~ unique namespace shall be assigned to each version of the profile schema.

Additional information that could be included in the profile schema:

- a~~A~~ description about the profile schema, i.e. the purpose of the profile, how to use it, when to use it etc;~~.~~
- a~~A~~ description of extensibility,i.e.how to handle future extensions of the profile schema.

A device capability profile can use an arbitrary number of vocabularies and thus it is possible to reuse attributes from other vocabularies by simply referencing the corresponding namespaces. The focus of the PSS vocabulary is content formatting which overlaps the focus of the UAProf vocabulary. UAProf is specified by WAP Forum and is an architecture and vocabulary/schema for capability exchange in the WAP environment. Since there are attributes in the UAProf vocabulary suitable for streaming applications these are reused and combined with a PSS application specific streaming component. This makes the PSS vocabulary an extension vocabulary to UAProf. The CC/PP specification encourages reuse of attributes from other vocabularies. To avoid confusion, the same attribute name should not be used in different vocabularies. In clause 5.2.3.3 a number of attributes from UAProf [40] are recommended for PSS. The PSS base vocabulary is defined in clause 5.2.3.2.

A profile is allowed to instantiate a subset of the attributes in the vocabularies and no specific attributes are required but insufficient description may lead to content unable to be shown by the client.

A.4.5 Principles of extending a schema/vocabulary

The use of RDF enables an extensibility mechanism for CC/PP-based schemas that addresses the evolution of new types of devices and applications. The PSS profile schema specification is going to provide a base vocabulary but in the future new usage scenarios might have need for expressing new attributes. This is the reason why there is a need to specify how extensions of the schema will be handled. If the TSG responsible for the present document updates the base vocabulary schema a new unique namespace will be assigned to the updated schema. In another scenario the TSG may decide to add a new component containing specific user related attributes. This new component will be assigned a new namespace and it will not influence the base vocabulary in any way. If other organisations or companies make extensions this can be either as a new component or as attributes added to the existing base vocabulary component where the new attributes uses a new namespace. This ensures that third parties can define and maintain their own vocabularies independently from the PSS base vocabulary.

A.4.6 Signalling of profile information between client and server

URLdesc and Profdiff were introduced in clause A.4.1. The URLdesc is a list of URLs that point to locations on device profile servers from where the PSS server retrieves suitable device capability profiles. The Profdiff contains additional capability description information; e.g. overrides for certain attribute values. Both URLdesc and Profdiff are encapsulated in RTSP and HTTP messages using additional header fields. This can be seen in Figure A.1. In clause 9.1 of [40] three new HTTP headers are defined that can be used to implement the desired functionality: "x-wap-profile", "x-wap-profile-diff" and "x-wap-profile-warning". These headers are reused in PSS for both HTTP and RTSP.

- The "x-wap-profile" is a request header that contains a list of absolute URLs to device capability descriptions and profile diff names. The profile diff names correspond to additional profile information in the "x-wap-profile-diff" header.
- The "x-wap-profile-diff" is a request header that contains a subset of a device capability profile.
- The "x-wap-profile-warning" is a response header that contains error codes explaining to what extent the server has been able to match the terminal request.

Clause 5.2.5 of the present document defines this exchange mechanism.

It is left to the mobile terminal to decide when to send x-wap-profile headers. The mobile terminal could send the "x-wap-profile" and "x-wap-profile-diff" headers with each RTSP DESCRIBE and/or with each RTSP SETUP request. Sending them in the RTSP DESCRIBE request is useful for the PSS server to be able to make a better decision which presentation description to provision to the client. Sending the "x-wap-profile" and "x-wap-profile-diff" headers with an HTTP request is useful whenever the mobile terminal requests some multimedia content that will be used in the PSS application. For example it can be sent with the request for a SMIL file and the PSS server can see to it that the mobile terminal receives a SMIL file which is optimised for the particular terminal. Clause 5.2.5 of the present document gives recommendations for when profile information should be sent.

It is up to the PSS server to retrieve the device capability profiles using the URLs in the "x-wap-profile" header. The PSS server is also responsible to merge the profiles then received. If the "x-wap-profile-diff" header is present it must also merge that information with the retrieved profiles. This functionality is defined in clause 5.2.6.

It should be noted that it is up the implementation of the mobile terminal what URLs to send in the "x-wap-profile" header. For instance, a terminal could just send one URL that points to a complete description of its capabilities. Another terminal might provide one URL that points to a description of the terminal hardware. A second URL that points to a description of a particular software version of the streaming application, and a third URL that points to the description of a hardware or software plug-in that is currently added to the standard configuration of that terminal. From this example it becomes clear that sending URLs from the mobile terminal to the server is good enough not only for static profiles but that it can also handle re-configurations of the mobile terminal such as software version changes, software plug-ins, hardware upgrades, etc.

As described above the list of URLs in the x-wap-profile header is a powerful tool to handle dynamic changes of the mobile terminal. The "x-wap-profile-diff" header could also be used to facilitate the same functionality. To use the "x-wap-profile-diff" header to e.g. send a complete profile (no URL present at all in the "x-wap-profile header") or updates as a result of e.g. a hardware plug-in is not recommended unless some compression scheme is applied over the air-interface. The reason is of course that the size of a profile may be large.

A.4.7 Example of a PSS device capability description

The following is an example of a device capability profile as it could be available from a device profile server. The XML document includes the description of the imaginary "Phone007" phone.

Instead of a single XML document the description could also be spread over several files. The PSS server would need to retrieve these profiles separately in this case and would need to merge them. For instance, this would be useful when device capabilities of this phone that are related to streaming would differ among different versions of the phone. In this case the part of the profile for streaming would be separated from the rest into its own profile document. This separation allows describing the difference in streaming capabilities by providing multiple versions of the profile document for the streaming capabilities.

```
<?xml version="1.0"?>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
```

```
         xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-ns"
         xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20010330"
         xmlns:pss6="http://www.3gpp.org/profiles/PSS/ccppschema-PSS6">

  <rdf:Description rdf:about="http://www.bar.com/Phones/Phone007">

    <ccpp:component>
      <rdf:Description ID="HardwarePlatform">
      <rdf:type rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010330#HardwarePlatform" />
        <prf:BitsPerPixel>4</prf:BitsPerPixel>
        <prf:ColorCapable>Yes</prf:ColorCapable>
        <prf:PixelAspectRatio>1x2</prf:PixelAspectRatio>
        <prf:PointingResolution>Pixel</prf:PointingResolution>

        <prf:Model>Phone007</prf:Model>
        <prf:Vendor>Ericsson</prf:Vendor>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID="SoftwarePlatform">
      <rdf:type rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010330#SoftwarePlatform" />
        <prf:CcppAccept-Charset>
          <rdf:Bag>
            <rdf:li>UTF-8</rdf:li>
            <rdf:li>ISO-10646-UCS-2</rdf:li>
          </rdf:Bag>
        </prf:CcppAccept-Charset>
        <prf:CcppAccept-Encoding>
          <rdf:Bag>
            <rdf:li>base64</rdf:li>
            <rdf:li>quoted-printable</rdf:li>
          </rdf:Bag>
        </prf:CcppAccept-Encoding>
        <prf:CcppAccept-Language>
          <rdf:Seq>
            <rdf:li>en</rdf:li>
        <rdf:li>se</rdf:li>
          </rdf:Seq>
        </prf:CcppAccept-Language>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID="PssCommon">
      <rdf:type rdf:resource="http://www.3gpp.org/profiles/PSS/ccppschema-PSS6#PssCommon" />
        <pss6:AudioChannels>Stereo</pss6:AudioChannels>
        <pss6:MaxPolyphony>24</pss6:MaxPolyphony>
        <pss6:PssVersion>3GPP-R6</pss6:PssVersion>
        <pss6:RenderingScreenSize>160x120</pss6:RenderingScreenSize>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID="Streaming">
      <rdf:type rdf:resource=" http://www.3gpp.org/profiles/PSS/ccppschema-PSS6#Streaming" />
        <pss6:3gppLinkChar>Yes</pss6:3gppLinkChar>
        <pss6:BufferFeedback>Yes</pss6:BufferFeedback>
        <pss6:ExtendedRtcpReports>Yes</pss6:ExtendedRtcpReports>
        <pss6:MediaAlternatives>Yes</pss6:3gppLinkChar>
        <pss6:RtpProfiles>
          <rdf:Bag>
            <rdf:li>RTP/AVP</rdf:li>
            <rdf:li>RTP/AVPF</rdf:li>
          </rdf:Bag>
        </pss6:RtpProfiles>
        <pss6:VideoPreDecoderBufferSize>30720</pss6:VideoPreDecoderBufferSize>
        <pss6:VideoInitialPostDecoderBufferingPeriod>0</pss6:VideoInitialPostDecoderBufferingPeriod>
        <pss6:VideoDecodingByteRate>16000</pss6:VideoDecodingByteRate>
        <pss6:StreamingAccept>
          <rdf:Bag>
            <rdf:li>audio/AMR</rdf:li>
            <rdf:li>audio/AMR-WB;octet-alignment=1</rdf:li>
            <rdf:li>video/H263-2000;profile=0;level=10</rdf:li>
            <rdf:li>video/H263-2000;profile=3;level=10</rdf:li>
            <rdf:li>video/MP4V-ES</rdf:li>
          </rdf:Bag>
        </pss6:StreamingAccept>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID="3gpFileFormat">
      <rdf:type rdf:resource=" http://www.3gpp.org/profiles/PSS/ccppschema-PSS6#3gpFileFormat" />
        <pss6:Brands>
          <rdf:Bag>
            <rdf:li>3gp4</rdf:li>
            <rdf:li>3gp5</rdf:li>
            <rdf:li>3gp6</rdf:li>
            <rdf:li>3gr6</rdf:li>
          </rdf:Bag>
        </pss6:Brands>
        <pss6:3gpAccept>
          <rdf:Bag>
            <rdf:li>audio/AMR</rdf:li>
            <rdf:li>audio/AMR-WB;octet-alignment=1</rdf:li>
            <rdf:li>video/H263-2000;profile=0;level=10</rdf:li>
            <rdf:li>video/H263-2000;profile=3;level=10</rdf:li>
            <rdf:li>video/Timed-Text</rdf:li>
          </rdf:Bag>
        </pss6:3gpAccept>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID="PssSmil">
      <rdf:type rdf:resource=" http://www.3gpp.org/profiles/PSS/ccppschema-PSS6#PssSmil" />
        <pss6:SmilAccept>
          <rdf:Bag>
            <rdf:li>Streaming-Media</rdf:li>
            <rdf:li>video/3gpp</rdf:li>
            <rdf:li>audio/AMR</rdf:li>
            <rdf:li>audio/sp-midi</rdf:li>
          </rdf:Bag>
        </pss6:SmilAccept>
        <pss6:SmilAccept-Subset>
          <rdf:Bag>
            <rdf:li>JPEG-PSS</rdf:li>
          </rdf:Bag>
        </pss6:SmilAccept-Subset>
        <pss6:SmilBaseSet>SMIL-3GPP-R6</pss6:SmilBaseSet>
        <pss6:SmilModules>
          <rdf:Bag>
            <rdf:li>BasicTransitions</rdf:li>
            <rdf:li>MulitArcTiming</rdf:li>
          </rdf:Bag>
        </pss6:SmilModules>
      </rdf:Description>
    </ccpp:component>

  </rdf:Description>
</rdf:RDF>
~~<?xml version="1.0"?>~~

~~<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"~~
         ~~xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp"~~
         ~~xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20010330"~~
         ~~xmlns:pss5="http://www.3gpp.org/profiles/PSS/ccppschema-PSS5">~~

  ~~<rdf:Description rdf:about="http://www.bar.com/Phones/Phone007">~~

    ~~<ccpp:component>~~
      ~~<rdf:Description ID="HardwarePlatform">~~
      ~~<rdf:type rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-~~
~~20010330#HardwarePlatform" />~~
        ~~<prf:BitsPerPixel>4</prf:BitsPerPixel>~~
        ~~<prf:ColorCapable>Yes</prf:ColorCapable>~~
        ~~<prf:PixelAspectRatio>1x2</prf:PixelAspectRatio>~~
        ~~<prf:PointingResolution>Pixel</prf:PointingResolution>~~

        ~~<prf:Model>Phone007</prf:Model>~~
        ~~<prf:Vendor>Ericsson</prf:Vendor>~~
      ~~</rdf:Description>~~
    ~~</ccpp:component>~~
    ~~<ccpp:component>~~
      ~~<rdf:Description ID="SoftwarePlatform">~~
      ~~<rdf:type rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-~~
~~20010330#SoftwarePlatform" />~~
        ~~<prf:CcppAccept-Charset>~~
          ~~<rdf:Bag>~~
            ~~<rdf:li>UTF-8</rdf:li>~~
            ~~<rdf:li>ISO-10646-UCS-2</rdf:li>~~
          ~~</rdf:Bag>~~
        ~~</prf:CcppAccept-Charset>~~
        ~~<prf:CcppAccept-Encoding>~~
          ~~<rdf:Bag>~~
            ~~<rdf:li>base64</rdf:li>~~
            ~~<rdf:li>quoted-printable</rdf:li>~~
          ~~</rdf:Bag>~~
        ~~</prf:CcppAccept-Encoding>~~
        ~~<prf:CcppAccept-Language>~~
          ~~<rdf:Seq>~~
            ~~<rdf:li>en</rdf:li>~~
        ~~<rdf:li>se</rdf:li>~~
          ~~</rdf:Seq>~~
        ~~</prf:CcppAccept-Language>~~
      ~~</rdf:Description>~~
    ~~</ccpp:component>~~

    ~~<ccpp:component>~~
      ~~<rdf:Description ID="Streaming">~~
      ~~<rdf:type rdf:resource=" http://www.3gpp.org/profiles/PSS/ccppschema-PSS5#Streaming" />~~
        ~~<pss5:AudioChannels>Stereo</pss5:AudioChannels>~~
        ~~<pss5:VideoPreDecoderBufferSize>30720</pss5:VideoPreDecoderBufferSize>~~
        ~~<pss5:VideoInitialPostDecoderBufferingPeriod>0</pss5:VideoInitialPostDecoderBufferingPeriod>~~
        ~~<pss5:VideoDecodingByteRate>16000</pss5:VideoDecodingByteRate>~~
        ~~<pss5:RenderingScreenSize>73x50</pss5:RenderingScreenSize>~~
    ~~<pss5:PssAccept>~~
          ~~<rdf:Bag>~~
            ~~<rdf:li>audio/AMR-WB;octet-alignment</rdf:li>~~
            ~~<rdf:li>video/MP4V-ES</rdf:li>~~
          ~~</rdf:Bag>~~
        ~~</pss5:PssAccept>~~
        ~~<pss5:PssAccept-Subset>~~
          ~~<rdf:Bag>~~
            ~~<rdf:li>JPEG-PSS</rdf:li>~~
          ~~</rdf:Bag>~~
        ~~</pss5:PssAccept-Subset>~~
        ~~<pss5:PssVersion>3GPP-R5</pss5:PssVersion>~~
        ~~<pss5:RenderingScreenSize>70x40</pss5:RenderingScreenSize>~~
        ~~<pss5:SmilBaseSet>SMIL-3GPP-R4</pss5:SmilBaseSet>~~
        ~~<pss5:SmilModules>~~
          ~~<rdf:Bag>~~
            ~~<rdf:li>BasicTransitions</rdf:li>~~
            ~~<rdf:li>MulitArcTiming</rdf:li>~~
          ~~</rdf:Bag>~~
        ~~</pss5:SmilModules>~~
      ~~</rdf:Description>~~
    ~~</ccpp:component>~~

  ~~</rdf:Description>~~
~~</rdf:RDF>~~
```

Annex B (informative): SMIL authoring guidelines

The SMIL authoring guidelines are given in [52].

~~B.1 General~~

~~This is an informative annex for SMIL presentation authors. Authors can expect that PSS clients can handle the SMIL module collection defined in clause 8.2, with the restrictions defined in this Annex. When creating SMIL documents the author is recommended to consider that terminals may have small displays and simple input devices. The media types and their encoding included in the presentation should be restricted to what is described in clause 7 of the present document. Considering that many mobile devices may have limited software and hardware capabilities, the number of media to be played simultaneous should be limited. For example, many devices will not be able to handle more than one video sequence at the time.~~

~~B.2 BasicLinking~~

~~The Linking Modules define elements and attributes for navigational hyperlinking, either through user interaction or through temporal events. The BasicLinking module defines the "a" and "area" elements for basic linking:~~

~~a Similar to the "a" element in HTML it provides a link from a media object through the href attribute (which contains the URI of the link's destination). The "a" element includes a number of attributes for defining the behaviour of the presentation when the link is followed.~~

~~area Whereas the a element only allows a link to be associated with a complete media object, the area element allows links to be associated with spatial and/or temporal portions of a media object.~~

~~The area element may be useful for enabling services that rely on interactivity where the display size is not big enough to allow the display of links alongside a media (e.g. QCIF video) window. Instead, the user could, for example, click on a watermark logo displayed in the video window to visit the company website.~~

~~Even if the area element may be useful some mobile terminals will not be able to handle area elements that include multiple selectable regions within an area element. One reason for this could be that the terminals do not have the appropriate user interface. Such area elements should therefore be avoided. Instead it is recommended that the "a" element be used. If the "area" element is used, the SMIL presentation should also include alternative links to navigate through the presentation; i.e. the author should not create presentations that rely on that the player can handle "area" elements.~~

~~B.3 BasicLayout~~

~~When defining the layout of a SMIL presentation, a content author needs to be aware that the targeted devices might have diverse properties that effect how the content can be rendered. The different sizes of the display area that can be used to render content on the targeted devices should be considered for defining the layout of the SMIL presentation. The root-layout window might represent the entire display or only parts of it.~~

~~Content authors are encouraged to create SMIL presentations that will work well with different resolutions of the rendering area. As mentioned in the SMIL2 recommendation content authors should use SMIL ContentControl functionality for defining multiple layouts for their SMIL presentation that are tailored to the specific needs of the whole range of targeted devices. Furthermore, authors should include a default layout (i.e. a layout determined by the SMIL player) that will be used when none of the author-defined layouts can be used.~~

~~Using relative position and size attributes in the definition of a region is also helpful for making SMIL presentations more portable across different display sizes; these features should also be used.~~

~~A 3GPP SMIL player should use the layout definition of a SMIL presentation for presenting the content whenever possible. When the SMIL player fails to use the layout information defined by the author it is free to present the content using a layout it determines by itself.~~

~~The "fit" attribute defines how different media should be fitted into their respective display regions.~~

~~The rendering and layout of some objects on a small display might be difficult and all mobile devices may not support features such as scroll bars. Therefore "fit=scroll" should not be used except for text content.~~

~~Due to hardware restrictions in mobile devices, operations such that scaling of a video sequence, or even images, may be very difficult to achieve. According to the SMIL 2.0 specification SMIL players may in these situations clip the content instead. To be sure of that the presentation is displayed as the author intended, video content should be encoded in a size suitable for the targeted terminals and it is recommended to use "fit=hidden".~~

~~B.4 EventTiming~~

~~The two attributes "endEvent" and "repeatEvent" in the EventTiming module may cause problems for a mobile SMIL player. The end of a media element triggers the "endEvent". In the same way the "repeatEvent" occurs when the second and subsequent iterations of a repeated element begin playback. Both these events rely on that the SMIL player receives information about that the media element has ended. One example could be when the end of a video sequence initiates the event. If the player has not received explicit information about the duration of the video sequence, e.g. by the "dur" attribute in SMIL or by some external source as the "a=range" field in SDP. The player will have to rely on the RTCP BYE message to decide when the video sequence ends. If the RTCP BYE message is lost, the player will have problems initiate the event. For these reasons is recommended that the "endEvent" and "repeatEvent" attributes are used with care, and if used the player should be provided with some additional information about the duration of the media element that triggers the event. This additional information could e.g. be the "dur" attribute in SMIL or the "a=range" field in SDP.~~

~~The "inBoundsEvent" and "outOfBoundsEvent" attributes assume that the terminal has a pointer device for moving the focus to within a window (i.e. clicking within a window). Not all terminals will support this functionality since they do not have the appropriate user interface. Hence care should be taken in using these particular event triggers.~~

~~B.5 MetaInformation~~

~~Authors are encouraged to make use of meta data whenever providing such information to the mobile terminal appears to be useful. However, they should keep in mind that some mobile terminals will parse but not process the meta data.~~

~~Furthermore, authors should keep in mind that excessive use of meta data will substantially increase the file size of the SMIL presentation that needs to be transferred to the mobile terminal. This may result in longer set-up times.~~

~~B.6 XML entities~~

~~Entities are a mechanism to insert XML fragments inside an XML document. Entities can be internal, essentially a macro expansion, or external. Use of XML entities in SMIL presentations is not recommended, as many current XML parsers do not fully support them.~~

~~B.7 XHTML Mobile Profile~~

~~When rendering texts in a SMIL presentation, authors are able to use XHTML Mobile Profile [47] that contains thirteen modules. However, some of the modules include non-text information. When referring to an XHTML Mobile Profile document from a SMIL document, authors should use only the required XHTML Host Language modules : Structure Module, Text Module, Hypertext Module and List Module. The use of the Image Module, in particular, should not be used. Images and other non-text contents should be included in the SMIL document.~~

~~NOTE: An XHTML file including a module which is not part of the XHTML Host Language modules may not be shown as intended. Also, an XHTML file which uses elements or attributes from the required XHTML Host Language modules and which uses elements or attributes that are not included in XHTML Basic Profile [28], may not render correctly on legacy handsets which implement only XHTML Basic. These are:~~

- ~~The start attribute on the 'ol' element in the List module~~
- ~~The value attribute on the 'li' element in the List module~~
- ~~The 'b' element in the Presentation module~~
- ~~The 'big' element in the Presentation module~~
- ~~The 'hr' element in the Presentation module~~
- ~~The 'i' element in the Presentation module~~
- ~~The 'small' element in the Presentation module~~

Annex C (normative): MIME media types

C.1 (void)

~~MIME media type H263-2000~~

~~MIME media type name: video~~
~~MIME subtype name: H263-2000~~

~~Required parameters: None~~

~~Optional parameters:~~
~~profile: H.263 profile number, in the range 0 through 8, specifying the supported H.263 annexes/subparts.~~
~~level: Level of bitstream operation, in the range 0 through 99, specifying the level of computational complexity of the decoding process. When no profile and level parameters are specified, Baseline Profile (Profile 0) level 10 are the default values.~~

~~The profile and level specifications can be found in [23]. Note that the RTP payload format for H263-2000 is the same as for H263-1998 and is defined in [14], but additional annexes/subparts are specified along with the profiles and levels.~~

~~NOTE: The above text will be replaced with a reference to the RFC describing the H263-2000 MIME media type as soon as this becomes available.~~

C.2 MIME media type sp-midi

MIME media type name: audio
MIME subtype name: sp-midi

Required parameters: none

Optional parameters: none

NOTE: The above text will be replaced with a reference to the RFC describing the sp-midi MIME media type as soon as this becomes available.

C.3 MIME media type mobile-xmf

MIME media type name: audio
MIME subtype name: mobile-xmf

Required parameters: none

Optional parameters:

prl:
prl is a string (inside double quotation marks “”) containing the playback resources included in all Content Description MetaDataItems of the Mobile XMF file. The string contains two digit hexadecimal numbers representing data bytes from the Content Description Meta Data. The same resource is listed only once. A playback resource contains two parts: a prefix and data. If the file includes Playback Resource Lists such as [00h 01h 00h 02h] and [00h 01h 00h 03h], the corresponding prl is “000100020003” containing playback resources 01, 02, and 03 with the prefix 00.

minimum-pr:
minimum-pr is a string containing the Maximum Instantaneous Resource (MIR) values from the first row of all MIR Count Tables corresponding to the playback resources listed in prl. Only the largest value from the values

of the same resource is chosen. If the file includes first rows of MIR Count Tables such as [02h 00h] and [01h 01h] corresponding to the above Playback Resource Lists, the corresponding minimum-pr is "020001". (02 is the largest of 2 and 1, 00 is the largest of 0, and 01 is the largest of 1.) minimum-pr requires the use of prl and the values in minimum-pr must be in the same order as the resources in prl. minimum-pr is the most important of minimum-pr and total-pr, because it defines the minimum playback requirements.

total-pr:
total-pr is a string containing the MIR values from the last row of all MIR Count Tables corresponding to the playback resources listed in prl. Only the largest value from the values of the same resource is chosen. If the file includes last rows of MIR Count Tables such as [05h 02h] and [06h 01h] corresponding to the above Playback Resource Lists, the corresponding total-pr is "060201". (06 is the largest of 5 and 6, 02 is the largest of 2, and 01 is the largest of 1.) total-pr requires the use of prl and the values in total-pr must be in the same order as the resources in prl.

NOTE: The above text will be replaced with a reference to the RFC describing the mobile-xmf MIME media type as soon as this becomes available.

C.4 MIME media type mobile-dls

MIME media type name: audio
MIME subtype name: dls

Required parameters: none

Optional parameters:

dls-type:
A comma-separated list of the midi types that this content conforms to, with the following specified values: 0, 1, and 2 signify Downloadable Sounds Level 1.1 content, Downloadable Sounds Level 2.1 content, Mobile Downloadable Sound content, respectively. If the parameter is not specified the content is Downloadable Sound level 1.1 (0). Any unknown values SHALL be ignored.

NOTE: The above text will be replaced with a reference to the RFC describing the dls MIME media type as soon as this becomes available.

Annex D (normative): 3GP files – codecs and identification

The definition of the 3GPP file format, including codec registration and file identification, is given in [50]. The timed text format is defined in [51].

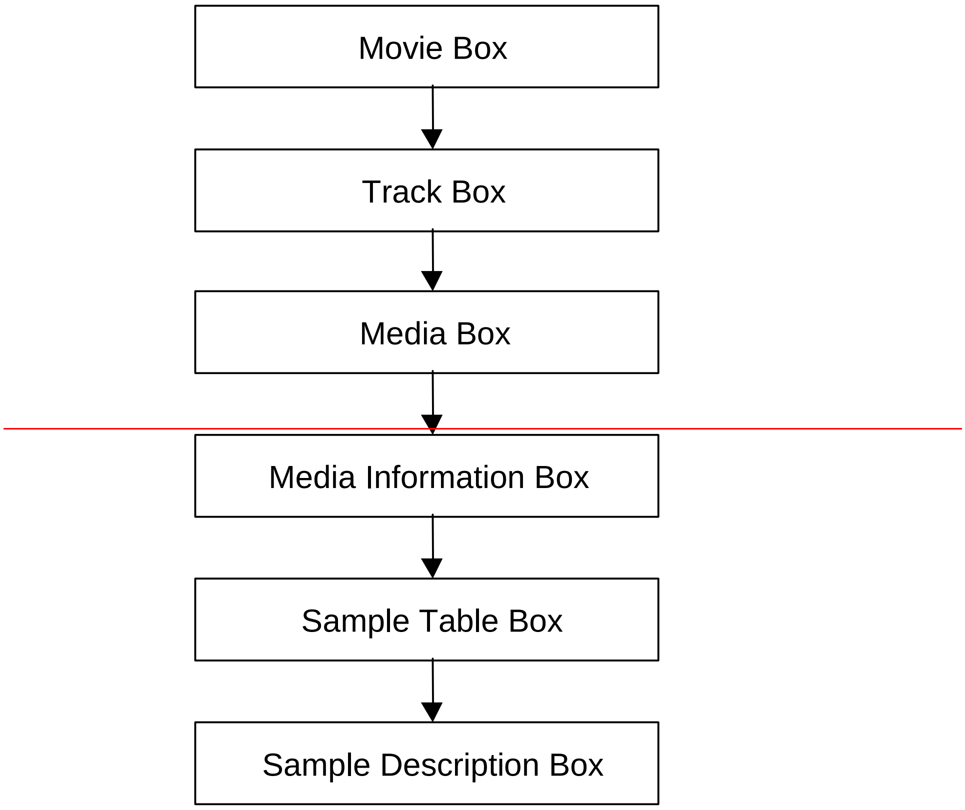
~~D.1 General~~

~~The purpose of this annex is to define the necessary structure for integration of the H.263, MPEG-4 video, AMR, AMR-WB, AAC and timed text media specific information in a 3GP file. Clause D.2 gives some background information about the Sample Description box in the ISO base media file format [50] and clauses D.3 and D.4 about the MP4VisualSampleEntry box and the MP4AudioSampleEntry box in the MPEG-4 file format [51]. The definitions of the SampleEntry boxes for AMR, AMR-WB and H.263 are given in clauses D.5 to D.8. The SampleEntry box for timed text is given in clause D.8a. Finally, the identification of 3GP files is described in clause D.9.~~

~~AMR and AMR-WB data is stored in the stream according to the AMR and AMR-WB storage format for single channel header of Annex E [11], without the AMR magic numbers.~~

~~D.2 Sample Description box~~

~~In an ISO file, Sample Description Box gives detailed information about the coding type used, and any initialisation information needed for that coding. The Sample Description Box can be found in the ISO file format Box Structure Hierarchy shown in figure D.1.~~



~~**Figure D.1: ISO File Format Box Structure Hierarchy**~~

~~The Sample Description Box can have one or more Sample Entries. Valid Sample Entries already defined for ISO and MP4 include MP4AudioSampleEntry, MP4VisualSampleEntry and HintSampleEntry. The Sample Entries for AMR and AMR-WB shall be AMRSampleEntry, for H.263 it shall be H263SampleEntry, and for timed text it shall be TextSampleEntry.~~

~~The format of SampleEntry and its fields are explained as follows:~~

~~SampleEntry ::= MP4VisualSampleEntry |~~

~~MP4AudioSampleEntry |~~

~~HintSampleEntry |~~

~~TextSampleEntry |~~

~~H263SampleEntry |~~

~~AMRSampleEntry~~

~~Table D.1: SampleEntry fields~~

~~Field~~	~~Type~~	~~Details~~	~~Value~~
~~MP4VisualSampleEntry~~		~~Entry type for visual samples defined in the MP4 specification.~~	
~~MP4AudioSampleEntry~~		~~Entry type for audio samples defined in the MP4 specification.~~	
~~HintSampleEntry~~		~~Entry type for hint track samples defined in the ISO specification.~~	
~~TextSampleEntry~~		~~Entry type for timed text samples defined in clause D8a.16 of the present document.~~	
~~H263SampleEntry~~		~~Entry type for H.263 visual samples defined in clause D.6 of the present document.~~	
~~AMRSampleEntry~~		~~Entry type for AMR and AMR-WB speech samples defined in clause D.5 of the present document.~~	

~~From the above 6 Sample Entries, only the MP4VisualSampleEntry, MP4AudioSampleEntry, TextSampleEntry, H263SampleEntry and AMRSampleEntry are taken into consideration, since hint tracks are out of the scope of the present document.~~

~~D.3 MP4VisualSampleEntry box~~

~~The MP4VisualSampleEntry Box is defined as follows:~~

~~MP4VisualSampleEntry ::= BoxHeader~~

~~Reserved_6~~

~~Data-reference-index~~

~~Reserved_16~~

~~Width~~

~~Height~~

~~Reserved_4~~

~~Reserved_4~~

~~Reserved_4~~

~~Reserved_2~~

~~Reserved_32~~

~~Reserved_2~~

~~Reserved_2~~

~~ESDBox~~

~~Table D.2: MP4VisualSampleEntry fields~~

~~Field~~	~~Type~~	~~Details~~	~~Value~~
~~**BoxHeader**.Size~~	~~Unsigned int(32)~~		
~~**BoxHeader**.Type~~	~~Unsigned int(32)~~		~~'mp4v'~~
~~Reserved_6~~	~~Unsigned int(8) [6]~~		~~0~~
~~Data-reference-index~~	~~Unsigned int(16)~~	~~Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference boxes.~~	
~~Reserved_16~~	~~Const unsigned int(32) [4]~~		~~0~~
~~Width~~	~~Unsigned int(16)~~	~~Maximum width, in pixels of the stream~~	
~~Height~~	~~Unsigned int(16)~~	~~Maximum height, in pixels of the stream~~	
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0x00480000~~
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0x00480000~~
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~1~~
~~Reserved_32~~	~~Const unsigned int(8) [32]~~		~~0~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~24~~
~~Reserved_2~~	~~Const int(16)~~		~~-1~~
~~**ESDBox**~~		~~Box containing an elementary stream descriptor for this stream.~~	

~~The stream type specific information is in the ESDBox structure, as defined in [51].~~

~~This version of the MP4VisualSampleEntry, with explicit width and height, shall be used for MPEG-4 video streams conformant to this specification.~~

~~NOTE: width and height parameters together may be used to allocate the necessary memory in the playback device without need to analyse the video stream.~~

~~D.4 MP4AudioSampleEntry box~~

~~MP4AudioSampleEntryBox is defined as follows:~~

~~MP4AudioSampleEntry ::= BoxHeader~~

~~Reserved_6~~

~~Data-reference-index~~

~~Reserved_8~~

~~Reserved_2~~

~~Reserved_2~~

~~Reserved_4~~

~~TimeScale~~

~~Reserved_2~~

~~**ESDBox**~~

~~Table D.3: MP4AudioSampleEntry fields~~

~~**Field**~~	~~**Type**~~	~~**Details**~~	~~**Value**~~
~~**BoxHeader**.Size~~	~~Unsigned int(32)~~		
~~**BoxHeader**.Type~~	~~Unsigned int(32)~~		~~'mp4a'~~
~~Reserved_6~~	~~Unsigned int(8) [6]~~		~~0~~
~~Data-reference-index~~	~~Unsigned int(16)~~	~~Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference boxes.~~	
~~Reserved_8~~	~~Const unsigned int(32) [2]~~		~~0~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~2~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~16~~
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0~~
~~TimeScale~~	~~Unsigned int(16)~~	~~Copied from track~~	
~~Reserved_2~~	~~Const unsigned int(16)~~		~~0~~
~~**ESDBox**~~		~~Box containing an elementary stream descriptor for this stream.~~	

~~The stream type specific information is in the ESDBox structure, as defined in [51].~~

~~D.5 AMRSampleEntry box~~

~~For narrow-band AMR, the box type of the AMRSampleEntry Box shall be 'samr'. For AMR wideband (AMR-WB), the box type of the AMRSampleEntry Box shall be 'sawb'.~~

~~The AMRSampleEntry Box is defined as follows:~~

~~**AMRSampleEntry ::= BoxHeader**~~

~~Reserved_6~~

~~Data-reference-index~~

~~Reserved_8~~

~~Reserved_2~~

~~Reserved_2~~

~~Reserved_4~~

~~TimeScale~~

~~Reserved_2~~

~~**AMRSpecificBox**~~

~~Table D.4: AMRSampleEntry fields~~

~~**Field**~~	~~**Type**~~	~~**Details**~~	~~**Value**~~
~~**BoxHeader**.Size~~	~~Unsigned int(32)~~		
~~**BoxHeader**.Type~~	~~Unsigned int(32)~~		~~'samr' or 'sawb'~~
~~Reserved_6~~	~~Unsigned int(8) [6]~~		~~0~~
~~Data-reference-index~~	~~Unsigned int(16)~~	~~Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference boxes.~~	
~~Reserved_8~~	~~Const unsigned int(32) [2]~~		~~0~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~2~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~16~~
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0~~
~~TimeScale~~	~~Unsigned int(16)~~	~~Copied from media header box of this media~~	
~~Reserved_2~~	~~Const unsigned int(16)~~		~~0~~
~~**AMRSpecificBox**~~		~~Information specific to the decoder.~~	

~~If one compares the MP4AudioSampleEntry Box - AMRSampleEntry Box the main difference is in the replacement of the ESDBox, which is specific to MPEG-4 systems, with a box suitable for AMR and AMR-WB. The **AMRSpecificBox** field structure is described in clause D.7.~~

~~D.6 H263SampleEntry box~~

~~The box type of the H263SampleEntry Box shall be 's263'.~~

~~The H263SampleEntry Box is defined as follows:~~

~~**H263SampleEntry ::= BoxHeader**~~

~~Reserved_6~~

~~Data-reference-index~~

~~Reserved_16~~

~~Width~~

~~Height~~

~~Reserved_4~~

~~Reserved_4~~

~~Reserved_4~~

~~Reserved_2~~

~~Reserved_32~~

~~Reserved_2~~

~~Reserved_2~~

~~**H263SpecificBox**~~

~~Table D.5: H263SampleEntry fields~~

~~**Field**~~	~~**Type**~~	~~**Details**~~	~~**Value**~~
~~**BoxHeader**.Size~~	~~Unsigned int(32)~~		
~~**BoxHeader**.Type~~	~~Unsigned int(32)~~		~~'s263'~~
~~Reserved_6~~	~~Unsigned int(8) [6]~~		~~0~~
~~Data-reference-index~~	~~Unsigned int(16)~~	~~Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference boxes.~~	
~~Reserved_16~~	~~Const unsigned int(32) [4]~~		~~0~~
~~Width~~	~~Unsigned int(16)~~	~~Maximum width, in pixels of the stream~~	
~~Height~~	~~Unsigned int(16)~~	~~Maximum height, in pixels of the stream~~	
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0x00480000~~
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0x00480000~~
~~Reserved_4~~	~~Const unsigned int(32)~~		~~0~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~1~~
~~Reserved_32~~	~~Const unsigned int(8) [32]~~		~~0~~
~~Reserved_2~~	~~Const unsigned int(16)~~		~~24~~
~~Reserved_2~~	~~Const int(16)~~		~~-1~~
~~**H263SpecificBox**~~		~~Information specific to the H.263 decoder.~~	

~~If one compares the MP4VisualSampleEntry – H263SampleEntry Box the main difference is in the replacement of the ESDBox, which is specific to MPEG-4 systems, with a box suitable for H.263. The **H263SpecificBox** field structure for H.263 is described in clause D.8.~~

~~D.7 AMRSpecificBox field for AMRSampleEntry box~~

~~The AMRSpecificBox fields for AMR and AMR-WB shall be as defined in table D.6. The AMRSpecificBox for the AMRSampleEntry Box shall always be included if the 3GP file contains AMR or AMR-WB media.~~

~~Table D.6: The AMRSpecificBox fields for AMRSampleEntry~~

~~Field~~	~~Type~~	~~Details~~	~~Value~~
~~BoxHeader.Size~~	~~Unsigned int(32)~~		
~~BoxHeader.Type~~	~~Unsigned int(32)~~		~~'damr'~~
~~DecSpecificInfo~~	~~AMRDecSpecStruc~~	~~Structure which holds the AMR and AMR-WB Specific information~~	

~~**BoxHeader Size and Type:** indicate the size and type of the AMR decoder-specific box. The type must be 'damr'.~~

~~**DecSpecificInfo:** the structure where the AMR and AMR-WB stream specific information resides.~~

~~The AMRDecSpecStruc is defined as follows:~~

~~*struct* **AMRDecSpecStruc**{~~

~~Unsigned int (32) **vendor**~~

~~Unsigned int (8) **decoder_version**~~

~~Unsigned int (16) **mode_set**~~

~~Unsigned int (8) **mode_change_period**~~

~~Unsigned int (8) **frames_per_sample**~~

~~}~~

~~The definitions of AMRDecSpecStruc members are as follows:~~

~~**vendor:** four character code of the manufacturer of the codec, e.g. 'VXYZ'. The vendor field gives information about the vendor whose codec is used to create the encoded data. It is an informative field which may be used by the decoding end. If a manufacturer already has a four character code, it is recommended that it uses the same code in this field. Else, it is recommended that the manufacturer creates a four character code which best addresses the manufacturer's name. It can be safely ignored.~~

~~**decoder_version:** version of the vendor's decoder which can decode the encoded stream in the best (i.e. optimal) way. This field is closely tied to the vendor field. It may give advantage to the vendor which has optimal encoder-decoder version pairs. The value is set to 0 if decoder version has no importance for the vendor. It can be safely ignored.~~

~~**mode_set:** the active codec modes. Each bit of the mode_set parameter corresponds to one mode. The bit index of the mode is calculated according to the 4 bit FT field of the AMR or AMR-WB frame structure. The mode_set bit structure is as follows: (B15xxxxxxB8B7xxxxxxB0) where B0 (Least Significant Bit) corresponds to Mode 0, and B8 corresponds to Mode 8.~~

~~The mapping of existing AMR modes to FT is given in table 1.a in [19]. A value of 0x81FF means all modes and comfort noise frames are possibly present in an AMR stream.~~

~~The mapping of existing AMR-WB modes to FT is given in Table 1.a in TS 26.201 [37]. A value of 0x83FF means all modes and comfort noise frames are possibly present in an AMR-WB stream.~~

~~As an example, if mode_set = 0000000110010101b, only Modes 0, 2, 4, 7 and 8 are present in the stream.~~

~~**mode_change_period:** defines a number N, which restricts the mode changes only at a multiple of N frames. If no restriction is applied, this value should be set to 0. If mode_change_period is not 0, the following restrictions apply to it according to the frames_per_sample field:~~

~~*if (mode_change_period < frames_per_sample)*~~

~~*frames_per_sample = k x (mode_change_period)*~~

~~*else if (mode_change_period > frames_per_sample)*~~

~~*mode_change_period = k x (frames_per_sample)*~~

~~*where k : integer [2, …]*~~

~~If mode_change_period is equal to frames_per_sample, then the mode is the same for all frames inside one sample.~~

~~**frames_per_sample:** defines the number of frames to be considered as 'one sample' inside the 3GP file. This number shall be greater than 0 and less than 16. A value of 1 means each frame is treated as one sample. A value of 10 means that 10 frames (of duration 20 msec each) are put together and treated as one sample. It must be noted that, in this case, one sample duration is 20 (msec/frame) x 10 (frame) = 200 msec. For the last sample of the stream, the number of frames can be smaller than frames_per_sample, if the number of remaining frames is smaller than frames_per_sample.~~

~~NOTE1: The "hinter", for the creation of the hint tracks, can use the information given by the AMRDecSpecStruc members.~~

~~NOTE2: The following AMR MIME parameters are not relevant to PSS: {mode_set, mode_change_period, mode_change_neighbor}. PSS servers should not send these parameters in SDP, and PSS clients shall ignore these parameters if received.~~

~~D.8 H263SpecificBox field for H263SampleEntry box~~

~~The H263SpecificBox fields for H. 263 shall be as defined in table D.7. The H263SpecificBox for the H263SampleEntry Box shall always be included if the 3GP file contains H.263 media.~~

~~The H263SpecificBox for H263 is composed of the following fields.~~

~~Table D.7: The H263SpecificBox fields H263SampleEntry~~

~~**Field**~~	~~**Type**~~	~~**Details**~~	~~**Value**~~
~~BoxHeader.Size~~	~~Unsigned int(32)~~		
~~BoxHeader.Type~~	~~Unsigned int(32)~~		~~'d263'~~
~~DecSpecificInfo~~	~~H263DecSpecStruc~~	~~Structure which holds the H.263 Specific information~~	
~~**BitrateBox**~~		~~Specific bitrate information (optional)~~	

~~**BoxHeader Size and Type:** indicate the size and type of the H.263 decoder-specific box. The type must be 'd263'.~~

~~**DecSpecificInfo:** This is the structure where the H263 stream specific information resides.~~

~~H263DecSpecStruc is defined as follows:~~

~~*struct* **H263DecSpecStruc**{~~

~~Unsigned int (32) **vendor**~~

~~Unsigned int (8) **decoder_version**~~

~~Unsigned int (8) **H263_Level**~~

~~Unsigned int (8) **H263_Profile**~~

~~}~~

~~The definitions of H263DecSpecStruc members are as follows:~~

~~**vendor:** four character code of the manufacturer of the codec, e.g. 'VXYZ'. The vendor field gives information about the vendor whose codec is used to create the encoded data. It is an informative field which may be used by the decoding end. If a manufacturer already has a four character code, it is recommended that it uses the same code in this field. Else,~~

~~it is recommended that the manufacturer creates a four character code which best addresses the manufacturer's name. It can be safely ignored.~~

~~**decoder_version:** version of the vendor's decoder which can decode the encoded stream in the best (i.e. optimal) way. This field is closely tied to the vendor field. It may give advantage to the vendor which has optimal encoder-decoder version pairs. . The value is set to 0 if decoder version has no importance for the vendor. It can be safely ignored.~~

~~**H263_Level and H263_Profile:** These two parameters define which H263 profile and level is used. These parameters are based on the MIME media type video/H263-2000. The profile and level specifications can be found in [23].~~

~~EXAMPLE 1: H.263 Baseline = {H263_Level = 10, H263_Profile = 0}~~

~~EXAMPLE 2: H.263 Profile 3 @ Level 10 = {H263_Level = 10 , H263_Profile = 3}~~

~~NOTE: The "hinter", for the creation of the hint tracks, can use the information given by the H263DecSpecStruc members.~~

~~The BitrateBox field shall be as defined in table D.7.1. The BitrateBox may be included if the 3GP file contains H.263 media.~~

~~The BitrateBox is composed of the following fields.~~

~~**Table D.7.1: The BitrateBox fields**~~

~~**Field**~~	~~**Type**~~	~~**Details**~~	~~**Value**~~
~~BoxHeader.Size~~	~~Unsigned int(32)~~		
~~BoxHeader.Type~~	~~Unsigned int(32)~~		~~'bitr'~~
~~DecBitrateInfo~~	~~DecBitrStruc~~	~~Structure which holds the Bitrate information~~	

~~**BoxHeader Size and Type:** indicate the size and type of the bitrate box. The type must be 'bitr'.~~

~~**DecBitrateInfo:** This is the structure where the stream bitrate information resides.~~

~~DecBitrStruc is defined as follows:~~

~~*struct* **DecBitrStruc**{~~

~~Unsigned int (32) **Avg_Bitrate**~~

~~Unsigned int (32) **Max_Bitrate**~~

~~}~~

~~The definitions of DecBitrStruc members are as follows:~~

~~**Avg_Bitrate:** the average bitrate in bits per second of this elementary stream. For streams with variable bitrate this value shall be set to zero.~~

~~**Max_Bitrate:** the maximum bitrate in bits per second of this elementary stream in any time window of one second duration.~~

~~D.8a Timed Text Format~~

~~This clause defines the format of timed text in downloaded files. In this release, timed text is downloaded, not streamed.~~

~~Operators may specify additional rules and restrictions when deploying terminals, in addition to this specification, and behavior that is optional here may be mandatory for particular deployments. In particular, the required character set is almost certainly dependent on the geography of the deployment.~~

~~D.8a.1 Unicode Support~~

~~Text in this specification uses the Unicode 3.0 [30] standard. Terminals shall correctly decode both UTF-8 and UTF-16 into the required characters. If a terminal receives a Unicode code, which it cannot display, it shall display a predictable result. It shall not treat multi-byte UTF-8 characters as a series of ASCII characters, for example.~~

~~Authors should create fully-composed Unicode; terminals are not required to handle decomposed sequences for which there is a fully-composed equivalent.~~

~~Terminals shall conform to the conformance statement in Unicode 3.0 section 3.1.~~

~~Text strings for display and font names are uniformly coded in UTF-8, or start with a UTF-16 BYTE ORDER MARK (\uFEFF) and by that indicate that the string which starts with the byte order mark is in UTF-16. Terminals shall recognise the byte-order mark in this byte order; they are not required to recognise byte-reversed UTF-16, indicated by a byte-reversed byte-order mark.~~

~~D.8a.2 Bytes, Characters, and Glyphs~~

~~This clause uses these terms carefully. Since multi-byte characters are permitted (i.e. 16-bit Unicode characters), the number of characters in a string may not be the number of bytes. Also, a byte-order-mark is not a character at all, though it occupies two bytes. So, for example, storage lengths are specified as byte-counts, whereas highlighting is specified using character offsets.~~

~~It should also be noted that in some writing systems the number of glyphs rendered might be different again. For example, in English, the characters 'fi' are sometimes rendered as a single ligature glyph.~~

~~In this specification, the first character is at offset 0 in the string. In records specifying both a start and end offset, the end offset shall be greater than or equal to the start offset. In cases where several offset specifications occur in sequence, the start offset of an element shall be greater than or equal to the end offset of the preceding element.~~

~~D.8a.3 Character Set Support~~

~~All terminals shall be able to render Unicode characters in these ranges:~~

~~a) basic ASCII and Latin-1 (\u0000 to \u00FF), though not all the control characters in this range are needed;~~

~~b) the Euro currency symbol (\u20AC)~~

~~c) telephone and ballot symbols (\u260E through \u2612)~~

~~Support for the following characters is recommended but not required:~~

~~a) miscellaneous technical symbols (\u2300 through \u2335)~~

~~b) 'Zapf Dingbats': locations \u2700 through \u27AF, and the locations where some symbols have been relocated (e.g. \u2605, Black star).~~

~~The private use characters \u0091 and \u0092, and the initial range of the private use area \uE000 through \uE0FF are reserved in this specification. For these Unicode values, and for control characters for which there is no defined graphical behaviour, the terminal shall not display any result: neither a glyph is shown nor is the current rendering position changed.~~

~~D.8a.4 Font Support~~

~~Fonts are specified in this specification by name, size, and style. There are three special names which shall be recognized by the terminal: Serif, Sans-Serif, and Monospace. It is strongly recommended that these be different fonts for the required characters from ASCII and Latin-1. For many other characters, the terminal may have a limited set or only a single font. Terminals requested to render a character where the selected font does not support that character should substitute a suitable font. This ensures that languages with only one font (e.g. Asian languages) or symbols for which there is only one form are rendered.~~

~~Fonts are requested by name, in an ordered list. Authors should normally specify one of the special names last in the list.~~

~~Terminals shall support a pixel size of 12 (on a 72dpi display, this would be a point size of 12). If a size is requested other than the size(s) supported by the terminal, the next smaller supported size should be used. If the requested size is smaller than the smallest supported size, the terminal should use the smallest supported size.~~

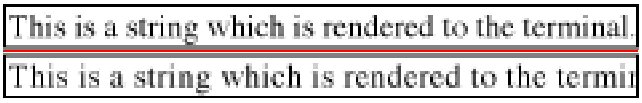
~~Terminals shall support unstyled text for those characters it supports. It may also support bold, italic (oblique) and bold-italic. If a style is requested which the terminal does not support, it should substitute a supported style; a character shall be rendered if the terminal has that character in any style of any font.~~

~~D.8a.5 Fonts and Metrics~~

~~Within the sample description, a complete list of the fonts used in the samples is found. This enables the terminal to pre-load them, or to decide on font substitution.~~

~~Terminals may use varying versions of the same font. For example, here is the same text rendered on two systems; it was authored on the first, where it just fitted into the text box.~~

~~**EXAMPLE:**~~



This is a string which is rendered to the terminal.

This is a string which is rendered to the termi

~~Authors should be aware of this possible variation, and provide text box areas with some ‘slack’ to allow for rendering variations.~~

~~D.8a.6 Colour Support~~

~~The colour of both text and background are indicated in this specification using RGB values. Terminals are not required to be able to display all colours in the RGB space. Terminals with a limited colour display, with only gray-scale display, and with only black-and-white are permissible. If a terminal has a limited colour capability it should substitute a suitable colour; dithering of text may be used but is not usually appropriate as it results in “fuzzy” display. If colour substitution is performed, the substitution shall be consistent: the same RGB colour shall result consistently in the same displayed colour. If the same colour is chosen for background and text, then the text shall be invisible (unless a style such as highlight changes its colour). If different colours are specified for the background and text, the terminal shall map these to different colours, so that the text is visible.~~

~~Colours in this specification also have an alpha or transparency value. In this specification, a transparency value of 0 indicates a fully transparent colour, and a value of 255 indicates fully opaque. Support for partial or full transparency is optional. ‘Keying’ text (text rendered on a transparent background) is done by using a background colour which is fully transparent. ‘Keying’ text over video or pictures, and support for transparency in general, can be complex and may require double-buffering, and its support is optional in the terminal. Content authors should beware that if they specify a colour which is not fully opaque, and the content is played on a terminal not supporting it, the affected area (the entire text box for a background colour) will be fully opaque and will obscure visual material behind it. Visual material with transparency is layered closer to the viewer than the material which it partially obscures.~~

~~D.8a.7 Text rendering position and composition~~

~~Text is rendered within a region (a concept derived from SMIL). There is a text box set within that region. This permits the terminal to position the text within the overall presentation, and also to render the text appropriately given the writing direction. For text written left to right, for example, the first character would be rendered at, or near, the left edge of the box, and with its baseline down from the top of the box by one baseline height (a value derived from the font and font size chosen). Similar considerations apply to the other writing directions.~~

~~Within the region, text is rendered within a text box. There is a default text box set, which can be over-ridden by a sample.~~

~~The text box is filled with the background colour; after that the text is painted in the text colour. If highlighting is requested one or both of these colours may vary.~~

~~Terminals may choose to anti-alias their text, or not.~~

~~The text region and layering are defined using structures from the ISO base media file format.~~

~~This track header box is used for text track:~~

```
~~aligned(8) class TrackHeaderBox~~
~~    extends FullBox('tkhd', version, flags){~~
~~    if (version==1) {~~
~~        unsigned int(64)    creation_time;~~
~~        unsigned int(64)    modification_time;~~
~~        unsigned int(32)    track_ID;~~
~~        const unsigned int(32)  reserved = 0;~~
~~        unsigned int(64)    duration;~~
~~    } else { // version==0~~
~~        unsigned int(32)    creation_time;~~
~~        unsigned int(32)    modification_time;~~
~~        unsigned int(32)    track_ID;~~
~~        const unsigned int(32)  reserved = 0;~~
~~        unsigned int(32)    duration;~~
~~    }~~
~~    const unsigned int(32)[2]   reserved = 0;~~
~~    int(16) layer;~~
~~    template int(16) alternate_group = 0;~~
~~    template int(16)    volume = 0;~~
~~    const unsigned int(16)  reserved = 0;~~
~~    template int(32)[9]     matrix=~~
~~        { 0x00010000,0,0,0,0x00010000,0,tx,ty,0x40000000 };~~
~~        // unity matrix~~
~~    unsigned int(32) width;~~
~~    unsigned int(32) height;~~
~~}~~
```

~~Visually composed tracks including video and text are layered using the 'layer' value. This compares, for example, to z-index in SMIL. More negative layer values are towards the viewer. (This definition is compatible with that in ISO/MJ2).~~

~~The region is defined by the track width and height, and translation offset. This corresponds to the SMIL region. The width and height are stored in the track header fields above. The sample description sets a text box within the region, which can be over-ridden by the samples.~~

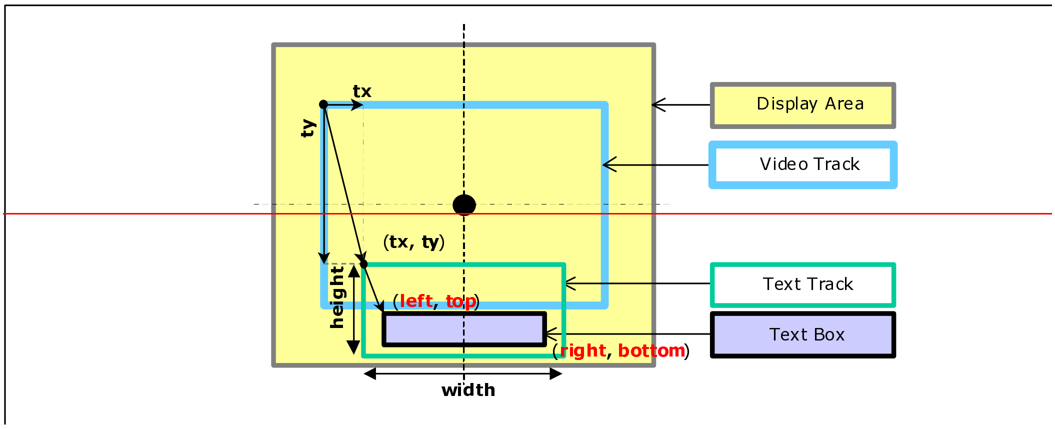
~~The translation values are stored in the track header matrix in the following positions:~~

~~{ 0x00010000,0,0, 0,0x00010000,0, tx, ty, 0x40000000 }~~

~~These values are fixed-point 16.16 values, here restricted to be integers (the lower 16 bits of each value shall be zero). The X axis increases from left to right; the Y axis from top to bottom. (This use of the matrix is conformant with ISO/MJ2.)~~

~~So, for example, a centered region of size 200x20, positioned below a video of size 320x240, would have track_width set to 200 (widh= 0x00c80000), track_height set to 20 (height= 0x00140000), and tx = (320-200)/2 = 60, and ty=240.~~

~~Since matrices are not used on the video tracks, all video tracks are set at the coordinate origin. Figure D.2 provides an overview:~~



~~Figure D.2: Illustration of text rendering position and composition~~

~~The top and left positions of the text track is determined by the tx and ty, which are the translation values from the coordinate origin (since the video track is at the origin, this is also the offset from the video track). The default text box set in the sample description sets the rendering area unless over-ridden by a 'tbox' in the text sample. The box values are defined as the relative values from the top and left positions of the text track.~~

~~It should be noted that this only specifies the relationship of the tracks within a single 3GP file. If a SMIL presentation lays up multiple files, their relative position is set by the SMIL regions. Each file is assigned to a region, and then within those regions the spatial relationship of the tracks is defined.~~

~~D.8a.8 Marquee Scrolling~~

~~Text can be 'marquee' scrolled in this specification (compare this to Internet Explorer's marquee construction). When scrolling is performed, the terminal first calculates the position in which the text would be displayed with no scrolling requested. Then:~~

~~a) If scroll-in is requested, the text is initially invisible, just outside the text box, and enters the box in the indicated direction, scrolling until it is in the normal position;~~

~~b) If scroll-out is requested, the text scrolls from the normal position, in the indicated direction, until it is completely outside the text box.~~

~~The rendered text is clipped to the text box in each display position, as always. This means that it is possible to scroll a string which is longer than can fit into the text box, progressively disclosing it (for example, like a ticker-tape). Note that both scroll in and scroll out may be specified; the text scrolls continuously from its invisible initial position, through the normal position, and out to its final position.~~

~~If a scroll-delay is specified, the text stays steady in its normal position (not initial position) for the duration of the delay; so the delay is after a scroll-in but before a scroll-out. This means that the scrolling is not continuous if both are specified. So without a delay, the text is in motion for the duration of the sample. For a scroll in, it reaches its normal position at the end of the sample duration; with a delay, it reaches its normal position before the end of the sample duration, and remains in its normal position for the delay duration, which ends at the end of the sample duration. Similarly for a scroll out, the delay happens in its normal position before scrolling starts. If both scroll in, and scroll out are specified, with a delay, the text scrolls in, stays stationary at the normal position for the delay period, and then scrolls out – all within the sample duration.~~

~~The speed of scrolling is calculated so that the complete operation takes place within the duration of the sample. Therefore the scrolling has to occur within the time left after scroll-delay has been subtracted from the sample duration. Note that the time it takes to scroll a string may depend on the rendered length of the actual text string. Authors should consider whether the scrolling speed that results will be exceed that at which text on a wireless terminal could be readable.~~

~~Terminals may use simple algorithms to determine the actual scroll speed. For example, the speed may be determined by moving the text an integer number of pixels in every update cycle. Terminals should choose a scroll speed which is as fast or faster than needed so that the scroll operation completes within the sample duration.~~

~~Terminals are not required to handle dynamic or stylistic effects such as highlight, dynamic highlight, or href links on scrolled text.~~

~~The scrolling direction is set by a two-bit field, with the following possible values:~~

~~00b – text is vertically scrolled up (‘credits style’), entering from the bottom of the bottom and leaving towards the top.~~

~~01b – text is horizontally scrolled (‘marquee style’), entering from the right and leaving towards the left.~~

~~10b – text is vertically scrolled down, entering from the top and leaving towards the bottom.~~

~~11b – text is horizontally scrolled, entering from the left and leaving towards the right.~~

~~D.8a.9 Language~~

~~The human language used in this stream is declared by the language field of the media-header box in this track. It is an ISO 639/T 3-letter code. The knowledge of the language used might assist searching, or speaking the text. Rendering is language neutral. Note that the values ‘und’ (undetermined) and ‘mul’ (multiple languages) might occur.~~

~~D.8a.10Writing direction~~

~~Writing direction specifies the way in which the character position changes after each character is rendered. It also will imply a start-point for the rendering within the box.~~

~~Terminals shall support the determination of writing direction, for those characters they support, according to the Unicode 3.0 specification. Note that the only required characters can all be rendered using left-right behaviour. A terminal which supports characters with right-left writing direction shall support the right-left composition rules specified in Unicode.~~

~~Terminals may also set, or allow the user to set, an overall writing direction, either explicitly or implicitly (e.g. by the language selection). This affects layout. For example, if upper-case letters are left-right, and lower-case right-left, and the Unicode string ABCdefGHI shall be rendered, it would appear as ABCfedGHI on a terminal with overall left-right writing (English, for example) and GHIfedABC on a system with overall right-left (Hebrew, for example).~~

~~Terminals are not required to support the bi-directional ordering codes (\u200E, \u200F and \u202A through \u202E).~~

~~If vertical text is requested by the content author, characters are laid out vertically from top to bottom. The terminal may choose to render different glyphs for this writing direction (e.g. a horizontal parenthesis), but in general the glyphs should not be rotated. The direction in which lines advance (left-right, as used for European languages, or right-left, as used for Asian languages) is set by the terminal, possibly by a direct or indirect user preference (e.g. a language setting). Terminals shall support vertical writing of the required character set. It is recommended that terminals support vertical writing of text in those languages commonly written vertically (e.g. Asian languages). If vertical text is requested for characters which the terminal cannot render vertically, the terminal may behave as if the characters were not available.~~

~~D.8a.11Text wrap~~

~~Automatic wrapping of text from line to line is complex, and can require hyphenation rules and other complex language-specific criteria. For these reasons, text is not wrapped in this specification. If a string is too long to be drawn within the box, it is clipped. The terminal may choose whether to clip at the pixel boundary, or to render only whole glyphs.~~

~~There may be multiple lines of text in a sample (hard wrap). Terminals shall start a new line for the Unicode characters line separator (\u2028), paragraph separator (\u2029) and line feed (\u000A). It is recommended that terminals follow Unicode Technical Report 13 [48]. Terminals should treat carriage return (\u000D), next line (\u0085) and CR+LF (\u000D\u000A) as new line.~~

~~D.8a.12Highlighting, Closed Caption, and Karaoke~~

~~Text may be highlighted for emphasis. Since this is a non-interactive system, solely for text display, the utility of this function may be limited.~~

~~Dynamic highlighting used for Closed Caption and Karaoke highlighting, is an extension of highlighting. Successive contiguous sub-strings of the text sample are highlighted at the specified times.~~

~~D.8a.13Media Handler~~

~~A text stream is its own unique stream type. For the 3GPP file format, the handler-type within the 'hdlr' box shall be 'text'.~~

~~D.8a.14Media Handler Header~~

~~The 3G text track uses an empty null media header ('nmhd'), called Mpeg4MediaHeaderBox in the MP4 specification [51], in common with other MPEG streams.~~

```
~~aligned(8) class  Mpeg4MediaHeaderBox~~
~~    extends FullBox('nmhd', version = 0, flags) {~~
~~ }~~
```

~~D.8a.15Style record~~

~~Both the sample format and the sample description contain style records, and so it is defined once here for compactness.~~

```
~~aligned(8) class StyleRecord {~~
~~    unsigned int(16)    startChar;~~
~~    unsigned int(16)    endChar;~~
~~    unsigned int(16)    font-ID;~~
~~    unsigned int(8) face-style-flags;~~
~~    unsigned int(8) font-size;~~
~~    unsigned int(8) text-color-rgba[4];~~
~~}~~
```

~~startChar: character offset of the beginning of this style run (always 0 in a sample description)~~

~~endChar: first character offset to which this style does not apply (always 0 in a sample description); shall be greater than or equal to startChar. All characters, including line-break characters and any other non-printing characters, are included in the character counts.~~

~~font-ID: font identifier from the font table; in a sample description, this is the default font~~

~~face style flags: in the absence of any bits set, the text is plain~~

~~1 bold~~

~~2 italic~~

~~4 underline~~

~~font-size: font size (nominal pixel size, in essentially the same units as the width and height)~~

~~text-color-rgba: rgb colour, 8 bits each of red, green, blue, and an alpha (transparency) value~~

~~Terminals shall support plain text, and underlined horizontal text, and may support bold, italic and bold-italic depending on their capabilities and the font selected. If a style is not supported, the text shall still be rendered in the closest style available.~~

~~D.8a.16Sample Description Format~~

~~The sample table box ('stbl') contains sample descriptions for the text track. Each entry is a sample entry box of type 'tx3g'. This name defines the format both of the sample description and the samples associated with that sample~~

~~description. Terminals shall not attempt to decode or display sample descriptions with unrecognised names, nor the samples attached to those sample descriptions.~~

~~It starts with the standard fields (the reserved bytes and the data reference index), and then some text-specific fields. Some fields can be overridden or supplemented by additional boxes within the text sample itself. These are discussed below.~~

~~There can be multiple text sample descriptions in the sample table. If the overall text characteristics do not change from one sample to the next, the same sample description is used. Otherwise, a new sample description is added to the table. Not all changes to text characteristics require a new sample description, however. Some characteristics, such as font size, can be overridden on a character-by-character basis. Some, such as dynamic highlighting, are not part of the text sample description and can be changed dynamically.~~

~~The TextDescription extends the regular sample entry with the following fields.~~

```
~~class FontRecord {~~
~~    unsigned int(16)    font-ID;~~
~~    unsigned int(8) font-name-length;~~
~~    unsigned int(8) font[font-name-length];~~
~~}~~

~~class FontTableBox() extends Box('ftab') {~~
~~    unsigned int(16) entry-count;~~
~~    FontRecord  font-entry[entry-count];~~
~~}~~

~~class BoxRecord {~~
~~    signed int(16)  top;~~
~~    signed int(16)  left;~~
~~    signed int(16)  bottom;~~
~~    signed int(16)  right;~~
~~}~~

~~class TextSampleEntry() extends SampleEntry ('tx3g') {~~
~~    unsigned int(32)    displayFlags;~~
~~    signed int(8)       horizontal-justification;~~
~~    signed int(8)       vertical-justification;~~
~~    unsigned int(8) background-color-rgba[4];~~
~~    BoxRecord           default-text-box;~~
~~    StyleRecord         default-style;~~
~~    FontTableBox        font-table;~~
~~}~~
```

~~displayFlags:~~
~~scroll In 0x00000020~~
~~scroll Out 0x00000040~~
~~scroll direction 0x00000180 / see above for values~~
~~continuous karaoke 0x00000800~~
~~write text vertically 0x00020000~~

~~horizontal and vertical justification: / two eight-bit values from the following list:~~
~~left, top 0~~
~~centered 1~~

~~bottom, right -1~~

~~background-color-rgba:~~
~~rgb color, 8 bits each of red, green, blue, and an alpha (transparency) value~~

~~Default text box: the default text box is set by four values, relative to the text region; it may be over-ridden in samples;~~

~~style record of default style: startChar and endChar shall be zero in a sample description~~

~~The text box is inset within the region defined by the track translation offset, width, and height. The values in the box are relative to the track region, and are uniformly coded with respect to the pixel grid. So, for example, the default text box for a track at the top left of the track region and 50 pixels high and 100 pixels wide is {0, 0, 50, 100}.~~

~~A font table shall follow these fields, to define the complete set of fonts used. The font table is a box of type ‘ftab’. Every font used in the samples is defined here by name. Each entry consists of a 16-bit local font identifier, and a font name, expressed as a string, preceded by an 8-bit field giving the length of the string in bytes. The name is expressed in UTF-8 characters, unless preceded by a UTF-16 byte-order-mark, whereupon the rest of the string is in 16-bit Unicode characters. The string should be a comma separated list of font names to be used as alternative font, in preference order. The special names “Serif”, “Sans-serif” and “Monospace” may be used. The terminal should use the first font in the list which it can support; if it cannot support any for a given character, but it has a font which can, it should use that font. Note that this substitution is technically character by character, but terminals are encouraged to keep runs of characters in a consistent font where possible.~~

~~D.8a.17 Sample Format~~

~~Each sample in the media data consists of a string of text, optionally followed by sample modifier boxes.~~

~~For example, if one word in the sample has a different size than the others, a 'styl' box is appended to that sample, specifying a new text style for those characters, and for the remaining characters in the sample. This overrides the style in the sample description. These boxes are present only if they are needed. If all text conforms to the sample description, and no characteristics are applied that the sample description does not cover, no boxes are inserted into the sample data.~~

```
~~class TextSampleModifierBox(type) extends Box(type) {~~
~~}~~

~~class TextSample {~~
~~    unsigned int(16)            text-length;~~
~~    unsigned int(8)         text[text-length];~~
~~    TextSampleModifierBox   text-modifier[];    // to end of the sample~~
~~}~~
```

~~The initial string is preceded by a 16-bit count of the number of bytes in the string. There is no need for null termination of the text string. The sample size table provides the complete byte-count of each sample, including the trailing modifier boxes; by comparing the string length and the sample size, you can determine how much space, if any, is left for modifier boxes.~~

~~Authors should limit the string in each text sample to not more than 2048 bytes, for maximum terminal interoperability.~~

~~Any unrecognised box found in the text sample should be skipped and ignored, and processing continue as if it were not there.~~

~~D.8a.17.1 Sample Modifier Boxes~~

~~D.8a.17.1.1 Text Style~~

~~'styl'~~

~~This specifies the style of the text. It consists of a series of style records as defined above, preceded by a 16-bit count of the number of style records. Each record specifies the starting and ending character positions of the text to which it applies. The styles shall be ordered by starting character offset, and the starting offset of one style record shall be greater than or equal to the ending character offset of the preceding record; styles records shall not overlap their character ranges.~~

```
~~class TextStyleBox() extends TextSampleModifierBox (‘styl’) {~~
~~    unsigned int(16)    entry-count;~~
~~    StyleRecord         text-styles[entry-count];~~
~~}~~
```

~~D.8a.17.1.2 Highlight~~

~~'hlit' - Specifies highlighted text: the box contains two 16-bit integers, the starting character to highlight, and the first character with no highlighting (e.g. values 4, 6 would highlight the two characters 4 and 5). The second value may be the number of characters in the text plus one, to indicate that the last character is highlighted.~~

```
~~class TextHighlightBox() extends TextSampleModifierBox (‘hlit’) {~~
~~    unsigned int(16)    startcharoffset;~~
~~    unsigned int(16)    endcharoffset;~~
~~}~~
~~class TextHilightColorBox() extends TextSampleModifierBox ('hclr') {~~
~~    unsigned int(8)     highlight_color_rgba[4];~~
~~}~~
```

~~highlight_color_rgb:~~

~~rgb color, 8 bits each of red, green, blue, and an alpha (transparency) value~~

~~The TextHilightColor Box may be present when the TextHighlightBox or TextKaraokeBox is present in a text sample. It is recommended that terminals use the following rules to determine the displayed effect when highlight is requested:~~

~~a) if a highlight colour is not specified, then the text is highlighted using a suitable technique such as inverse video: both the text colour and the background colour change.~~

~~b) if a highlight colour is specified, the background colour is set to the highlight colour for the highlighted characters; the text colour does not change.~~

~~Terminals do not need to handle text that is both scrolled and either statically or dynamically highlighted. Content authors should avoid specifying both scroll and highlight for the same sample.~~

~~D.8a.17.1.3 Dynamic Highlight~~

~~'krok' – Karaoke, closed caption, or dynamic highlighting. The number of highlight events is specified, and each event is specified by a starting and ending character offset and an end time for the event. The start time is either the sample start time or the end time of the previous event. The specified characters are highlighted from the previous end-time (initially the beginning of this sample’s time), to the end time. The times are all specified relative to the sample’s time; that is, a time of 0 represents the beginning of the sample time. The times are measured in the timescale of the track.~~

~~The box starts with the start-time offset of the first highlight event, a 16-bit count of the event count, and then that number of 8-byte records. Each record contains the end-time offset as a 32-bit number, and the text start and end values, each as a 16-bit number. These values are specified as in the highlight record – the offset of the first character to highlight, and the offset of the first character not highlighted. The special case, where the startcharoffset equals to the endcharoffset, can be used to pause during or at the beginning of dynamic highlighting. The records shall be ordered and not overlap, as in the highlight record. The time in each record is the end time of this highlight event; the first highlight event starts at the indicated start-time offset from the start time of the sample. The time values are in the units expressed by the timescale of the track. The time values shall not exceed the duration of the sample.~~

~~The continuouskaraoke flag controls whether to highlight only those characters (continuouskaraoke = 0) selected by a karaoke entry, or the entire string from the beginning up to the characters highlighted (continuouskaraoke = 1) at any given time. In other words, the flag specifies whether karaoke should ignore the starting offset and highlight all text from the beginning of the sample to the ending offset.~~

~~Karaoke highlighting is usually achieved by using the highlight colour as the text colour, without changing the background.~~

~~At most one dynamic highlight (‘krok’) box may occur in a sample.~~

```
~~class TextKaraokeBox() extends TextSampleModifierBox (‘krok’) {~~
~~    unsigned int(32)    highlight-start-time;~~
~~    unsigned int(16)    entry-count;~~
~~    for (i=1; i<=entry-count; i++) {~~
~~        unsigned int(32)    highlight-end-time;~~
~~        unsigned int(16)    startcharoffset;~~
~~        unsigned int(16)    endcharoffset;~~
~~    }~~
~~}~~
```

~~D.8a.17.1.4 Scroll Delay~~

~~'dlay' - Specifies a delay after a Scroll In and/or before Scroll Out. A 32-bit integer specifying the delay, in the units of the timescale of the track. The default delay, in the absence of this box, is 0.~~

```
~~class TextScrollDelayBox() extends TextSampleModifierBox (‘dlay’) {~~
~~    unsigned int(32)    scroll-delay;~~
~~}~~
```

~~D.8a.17.1.5 HyperText~~

~~'href' – HyperText link. The existence of the hypertext link is visually indicated in a suitable style (e.g. underlined blue text).~~

~~This box contains these values:~~

~~startCharOffset: – the start offset of the text to be linked~~

~~endCharOffset: – the end offset of the text (start offset + number of characters)~~

~~URLLength: - the number of bytes in the following URL~~

~~URL: UTF-8 characters – the linked-to URL~~

~~altLength: - the number of bytes in the following “alt” string~~

~~altstring: UTF-8 characters – an “alt” string for user display~~

~~The URL should be an absolute URL, as the context for a relative URL may not always be clear.~~

~~The “alt” string may be used as a tool-tip or other visual clue, as a substitute for the URL, if desired by the terminal, to display to the user as a hint on where the link refers.~~

~~Hypertext-linked text should not be scrolled; not all terminals can display this or manage the user interaction to determine whether user has interacted with moving text. It is also hard for the user to interact with scrolling text.~~

```
~~class TextHyperTextBox() extends TextSampleModifierBox (‘href’) {~~
~~    unsigned int(16)    startcharoffset;~~
~~    unsigned int(16)    endcharoffset;~~
~~    unsigned int(8) URLLength;~~
~~    unsigned int(8) URL[URLLength];~~
~~    unsigned int(8) altLength;~~
~~    unsigned int(8) altstring[altLength];~~
~~}~~
```

~~D.8a.17.1.6 Textbox~~

~~‘tbox’ – text box over-ride. This over-rides the default text box set in the sample description.~~

```
~~class TextboxBox() extends TextSampleModifierBox ('tbox') {~~
~~    BoxRecord   text-box;~~
~~}~~
```

~~D.8a.17.1.7 Blink~~

~~‘blnk’ – Blinking text. This requests blinking text for the indicated character range. Terminals are not required to support blinking text, and the precise way in which blinking is achieved, and its rate, is terminal-dependent.~~

```
~~class BlinkBox() extends TextSampleModifierBox ('blnk') {~~
~~    unsigned int(16)        startcharoffset;~~
~~    unsigned int(16)        endcharoffset;~~
~~}~~
```

~~D.8a.18 Combinations of features~~

~~Two modifier boxes of the same type shall not be applied to the same character (e.g. it is not permitted to have two href links from the same text). As the ‘hclr’, ‘dlay’ and ‘tbox’ are globally applied to the whole text in a sample, each sample shall contain at most one ‘hclr’, at most one ‘dlay’, and at most one ‘tbox’ modifier.~~

~~Table D.8 details the effects of multiple options:~~

~~**Table D.8: Combinations of features**~~

			~~First sample modifier box~~				
~~-~~	~~-~~	~~**Sample description style record**~~	~~**styl**~~	~~**hlit**~~	~~**krok**~~	~~**href**~~	~~**blnk**~~
~~**Second sample modifier box**~~	~~**styl**~~	~~1~~	~~3~~	~~-~~	~~-~~	~~-~~	~~-~~
	~~**hlit**~~	~~-~~	~~-~~	~~3~~	~~-~~	~~-~~	~~-~~
~~-~~	~~**krok**~~	~~-~~	~~-~~	~~4~~	~~3~~	~~-~~	~~-~~
~~-~~	~~**href**~~	~~2~~	~~2~~	~~-~~	~~5~~	~~3~~	~~-~~
~~**-**~~	~~**blnk**~~	~~-~~	~~6~~	~~6~~	~~6~~	~~6~~	~~6~~

1. ~~The sample description provides the default style; the style records over-ride this for the selected characters.~~

2. ~~The terminal over-rides the chosen style for HREF links.~~

3. ~~Two records of the same type cannot be applied to the same character.~~

4. ~~Dynamic and static highlighting must not be applied to the same text.~~

5. ~~Dynamic highlighting and linking must not be applied to the same text.~~

6. ~~Blinking text is optional, particularly when requested in combination with other features.~~

~~D.9 File Identification~~

~~3GPP multimedia files can be identified using several mechanisms. When stored in traditional computer file systems, these files should be given the file extension “.3gp” (readers should allow mixed case for the alphabetic characters). The MIME types “video/3gpp” (for visual or audio/visual content, where visual includes both video and timed text) and “audio/3gpp” (for purely audio content) are expected to be registered and used.~~

~~A file-type box, as defined in the ISO base media file format specification [50] shall be present in conforming files. The file type box ‘ftyp’ shall occur before any variable-length box (e.g. movie, free space, media data). Only a fixed-size box such as a file signature, if required, may precede it.~~

~~The brand identifier for this specification is '3gp5'. This brand identifier must occur in the compatible brands list, and may also be the primary brand. If the file is also conformant to release 4 of this specification, it is recommended that the Release 4 brand '3gp4' also occur in the compatible brands list; if 3gp4 is not in the compatible brand list the file will not be processed by a Release 4 reader. Readers should check the compatible brands list for the identifiers they recognize, and not rely on the file having a particular primary brand, for maximum compatibility. Files may be compatible with more than one brand, and have a 'best use' other than this specification, yet still be compatible with this specification.~~

~~**Table D.9: The File-Type box**~~

~~**Field**~~	~~**Type**~~	~~**Details**~~	~~**Value**~~
~~**BoxHeader**.Size~~	~~Unsigned int(32)~~		
~~**BoxHeader**.Type~~	~~Unsigned int(32)~~		~~'ftyp'~~
~~Brand~~	~~Unsigned int(32)~~	~~The major or 'best use' of this file~~	
~~MinorVersion~~	~~Unsigned int(32)~~		
~~CompatibleBrands~~	~~Unsigned int(32)~~	~~A list of brands, to end of the box~~	

~~**Brand**: Identifies the 'best use' of this file. The brand should match the file extension. For files with extension '.3gp' and conforming to this specification, the brand shall be '3gp5'.~~

~~**MinorVersion**: This identifies the minor version of the brand. For files with brand '3gpZ', where Z is a digit, and conforming to release Z.x.y, this field takes the value x*256 + y.~~

~~**CompatibleBrands**: a list of brand identifiers (to the end of the box). '3gp5' shall be a member of this list.~~

Annex E (normative): RTP payload format and file storage format for AMR and AMR-WB audio

The AMR and AMR-WB speech codec RTP payload, storage format and MIME type registration are specified in [11].

Annex F (normative): RDF schema for the PSS base vocabulary

```
<?xml version="1.0"?>

<!--
     This document is the RDF Schema for Packet-switched Streaming 
     Service (PSS)-specific vocabulary as defined in 3GPP TS 26.234 
     Release 6 (in the following "the specification").

     The URI for unique identification of this RDF Schema is 
       http://www.3gpp.org/profiles/PSS/ccppschema-PSS6

     This RDF Schema includes the same information as the respective
     chapter of the specification. Greatest care has been taken to keep
     the two documents consistence. However, in case of any divergence
     the specification takes presidence.

     All reference in this RDF Schmea are to be interpreted relative to
     the specification. This means all references using the form
     [ref] are defined in chapter 2 "References" of the specification. 
     All other references refer to parts within that document.

     Note: This Schemas has been aligned in structure and base
     vocabulary to the RDF Schema used by UAProf [40].

-->

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema" >

<!-- ****************************************************************** -->
<!-- ***** Properties shared among the components***** -->

  <rdf:Description ID="defaults">
    <rdfs:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:domain rdf:resource="#3gpFileFormat"/>
    <rdfs:domain rdf:resource="#PssSmil"/>
    <rdfs:comment>
      An attribute used to identify the default capabilities.
    </rdfs:comment>
  </rdf:Description>

<!-- ****************************************************************** -->
<!-- ***** Component Definitions ***** -->

  <rdf:Description ID="PssCommon">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010330#Component"/>
    <rdfs:label>Component: PssCommon</rdfs:label>
    <rdfs:comment>
      The PssCommon component specifies the base vocabulary common for all 
      PSS applications, in contrast to application-specific parts of the PSS 
      base vocabulary which are described by the Streaming, 3gpFileFormat and 
      PssSmil components defined below.

      PSS servers supporting capability exchange should understand the attributes
      in this component as explained in detail in 3GPP TS 26.234 Release 6..
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="Streaming">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010330#Component"/>
    <rdfs:label>Component: Streaming</rdfs:label>
    <rdfs:comment>
      The Streaming component specifies the base vocabulary for pure RTSP/RTP-
      based streaming in PSS.

      PSS servers supporting capability exchange should understand the attributes
      in this component as explained in detail in 3GPP TS 26.234 Release 6.
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="3gpFileFormat">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010330#Component"/>
    <rdfs:label>Component: 3gpFileFormat</rdfs:label>
    <rdfs:comment>
      The 3gpFileFormat component specifies the base vocabulary for 3GP file
      download or progressive download in PSS.

      PSS servers supporting capability exchange should understand the attributes
      in this component as explained in detail in 3GPP TS 26.234 Release 6.
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="PssSmil">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010330#Component"/>
    <rdfs:label>Component: PssSmil</rdfs:label>
    <rdfs:comment>
      The PssSmil component specifies the base vocabulary for SMIL presentations
      in PSS. Note that capabilities regarding streaming and 3GP files that are
      part of a SMIL presentation are expressed by the vocabularies specified by
      the Streaming and 3gpFileFormat components, respectively.

      PSS servers supporting capability exchange should understand the attributes
      in this component as explained in detail in 3GPP TS 26.234 Release 6.
    </rdfs:comment>
  </rdf:Description>

<!-- **
     ** In the following property definitions, the defined types
     ** are as follows:
     **
     ** Number: A positive integer
     ** [0-9]+
     ** Boolean: A yes or no value
     ** Yes|No
     ** Literal: An alphanumeric string
     ** [A-Za-z0-9/.\-_]+
     ** Dimension: A pair of numbers
     ** [0-9]+x[0-9]+
     **
-->

<!-- ****************************************************************** -->
<!-- ***** Component: PssCommon ***** -->

  <rdf:Description ID="AudioChannels">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: This attribute describes the stereophonic capability of the
      natural audio device. The only legal values are "Mono" and "Stereo".

      Type: Literal
      Resolution: Locked
      Examples: "Mono", "Stereo"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="MaxPolyphony">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: The MaxPolyphony attribute refers to the maximal polyphony
      that the synthetic audio device supports as defined in [44]. Legal values
      are integer between 5 to 24.
      NOTE: MaxPolyphony attribute can be used to signal the maximum polyphony
            capabilities supported by the PSS client. This is a complementary
            mechanism for the delivery of compatible SP-MIDI content and thus
            the PSS client is required to support Scalable Polyphony MIDI i.e.
            Channel Masking defined in [44].
      Type: Number
      Resolution: Locked
      Examples: 8
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="NumOfGM1Voices">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: The NumOfGM1Voices attribute refers to the maximum number
      of simultaneous GM1   voices that the synthetic audio engine supports.
      Legal values are integers greater or equal than 5.

      Type: Number
      Resolution: Locked
      Examples: 24
    </rdfs:comment>
  </rdf:Description>

<rdf:Description ID="NumOfMobileDLSVoicesWithoutOptionalBlocks">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: The NumOfMobileDLSVoicesWithoutOptionalBlocks attribute
      refers to the maximum number of simultaneous voices without optional
      group of processing blocks that the synthetic audio engine supports.
      Legal values are integers greater or equal than 5.

      Type: Number
      Resolution: Locked
      Examples: 24
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="NumOfMobileDLSVoicesWithOptionalBlocks">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: The NumOfMobileDLSVoicesWithOptionalBlocks attribute refers
      to the maximum number of simultaneous voices with optional group of
      processing blocks that the synthetic audio engine supports. This attribute
      is set to zero for devices that do not support the optional group of
      processing blocks. Legal values are integers greater or equal than 0.

      Type: Number
      Resolution: Locked
      Examples: 24
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="PssVersion">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: Latest PSS version supported by the client. Legal
      values are "3GPP-R4", "3GPP-R5", "3GPP-R6" and so forth.

      Type: Literal
      Resolution: Locked
      Examples: "3GPP-R5", "3GPP-R6"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="RenderingScreenSize">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssCommon"/>
    <rdfs:comment>
      Description: The rendering size of the device's screen in unit of
      pixels available for PSS media presentation. The horizontal size is
      given followed by the vertical size. Legal values are pairs of integer
      values equal or greater than zero. A value equal "0x0"means that there
      exists no display or just textual output is supported.

      Type: Dimension
      Resolution: Locked
      Examples: "160x120"
    </rdfs:comment>
  </rdf:Description>

<!-- ****************************************************************** -->
<!-- ***** Component: Streaming ***** -->

  <rdf:Description ID="StreamingAccept">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: List of content types (MIME types) relevant for streaming
      over RTP supported by the PSS application. Content types listed shall be
      possible to stream over RTP. For each content type a set of MIME parameters
      can be specified to signal receiver capabilities. A content type that
      supports multiple parameter sets may occur several times in the list.
      Legal values are lists of MIME types with related parameters.

      Type: Literal (bag)
      Resolution: Append
      Examples: "audio/AMR-WB;octet-alignment=1,application/smil"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="StreamingAccept-Subset">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: List of content types for which the PSS application supports
      a subset. MIME types can in most cases effectively be used to express
      variations in support for different media types. Many MIME types, e.g.
      AMR-WB has several parameters that can be used for this purpose. There
      may exist content types for which the PSS application only supports a
      subset and this subset cannot be expressed with MIME-type parameters.
      In these cases the attribute StreamingAccept-Subset is used to describe
      support for a subset of a specific content type. If a subset of a specific
      content type is declared in StreamingAccept-Subset, this means that
      StreamingAccept-Subset has precedence over StreamingAccept.
      StreamingAccept shall always include the corresponding content types for
      which StreamingAccept-Subset specifies subsets of.
      No legal values are currently defined.

      Type: Literal (bag)
      Resolution: Locked
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="3gppLinkChar">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: This attribute indicates whether the device supports the
      3GPP-Link-Char header according to clause 10.2.1.1 of the specification.
      Legal values are "Yes" and "No".

      Type: Number
      Resolution: Override
      Examples: "Yes"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="AdaptationSupport">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: This attribute indicates whether the device supports
      client buffer feedback signaling according to clause 10.2.3 of the
      specification. Legal values are "Yes" and "No".

      Type: Number
      Resolution: Locked
      Examples: "Yes"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="ExtendedRtcpReports">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: This attribute indicates whether the device supports
      extended RTCP reports according to clause 6.2.3.1 of the specification.
      Legal values are "Yes" and "No".

      Type: Number
      Resolution: Locked
      Examples: "Yes"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="MediaAlternatives">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: This attribute indicates whether the device interprets the
      SDP attributes "alt", "alt-default-id", and "alt-group", defined in
      clauses 5.3.3.3 and 5.3.3.4 of the specification.
      Legal values are "Yes" and "No".

      Type: Number
      Resolution: Override
      Examples: "Yes"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="RtpProfiles">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: This attribute lists the supported RTP profiles. Legal
      values are profile names registered through the Internet Assigned Numbers
      Authority (IANA), www.iana.org.

      Type: Literal (bag)
      Resolution: Append
      Examples: "RTP/AVP,RTP/AVPF"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="VideoDecodingByteRate ">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: If Annex G is not supported, the attribute has no meaning.
      If Annex G is supported, this attribute defines the peak decoding byte
      rate the PSS client is able to support. In other words, the PSS client
      fulfils the requirements given in Annex G with the signalled peak decoding
      byte rate. The values are given in bytes per second and shall be greater
      than or equal to 8000. According to Annex G, 8000 is the default peak
      decoding byte rate for the mandatory video codec profile and level
      (H.263 Profile 0 Level 10). Legal values are integer values greater than
      or equal to 8000.

      Type: Number
      Resolution: Locked
      Examples: "16000"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="VideoInitialPostDecoderBufferingPeriod">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: If Annex G is not supported, the attribute has no
      meaning. If Annex G is supported, this attribute defines the
      maximum initial post-decoder buffering period of video. Values are
      interpreted as clock ticks of a 90-kHz clock. In other words, the
      value is incremented by one for each 1/90 000 seconds. For
      example, the value 9000 corresponds to 1/10 of a second initial
      post-decodder buffering. Legal values are all integer values equal
      to or greater than zero.

      Type: Number
      Resolution: Locked
      Examples: "9000"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="VideoPreDecoderBufferSize">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#Streaming"/>
    <rdfs:comment>
      Description: This attribute signals if the optional video
      buffering requirements defined in Annex G are supported. It also
      defines the size of the hypothetical pre-decoder buffer defined in
      Annex G. A value equal to zero means that Annex G is not
      supported. A value equal to one means that Annex G is
      supported. In this case the size of the buffer is the default size
      defined in Annex G.  A value equal to or greater than the default
      buffer size defined in Annex G means that Annex G is supported and
      sets the buffer size to the given number of octets. Legal values are all
      integer values equal to or greater than zero. Values greater than
      one but less than the default buffer size defined in Annex G are
      not allowed.

      Type: Number
      Resolution: Locked
      Examples: "0", "4096"
    </rdfs:comment>
  </rdf:Description>

<!-- ****************************************************************** -->
<!-- ***** Component: 3gpFileFormat ***** -->

  <rdf:Description ID="Brands">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#3gpFileFormat"/>
    <rdfs:comment>
      Description: This attribute lists the supported 3GP profiles identified
      by brand. Legal values are brand identifiers according to 5.3.4 and 5.4
      in [50].

      Type: Literal (bag)
      Resolution: Append
      Examples: "3gp4,3gp5,3gp6,3gr6"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="3gpAccept">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#3gpFileFormat"/>
    <rdfs:comment>
      Description: List of content types (MIME types) that can be included
      in a 3GP file and handled by the PSS application. For each content
      type a set of supported parameters can be given. A content type that
      supports multiple parameter sets may occur several times in the list.
      A 3GP file may include timed text [51] and to declare support for this
      format an identifier ("Timed-Text") shall be used, since no MIME type
      exists. Legal values are lists of MIME types with related parameters
      and the "Timed-Text" identifier.

      Type: Literal (bag)
      Resolution: Append
      Examples: "video/H263-2000;profile=0;level=10,audio/AMR,Timed-text"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="3gpAccept-Subset">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#3gpFileFormat"/>
    <rdfs:comment>
      Description: List of content types for which the PSS application
      supports a subset. MIME types can in most cases effectively be used
      to express variations in support for different media types. Many MIME
      types have several parameters that can be used for this purpose. There
      may exist content types for which the PSS application only supports a
      subset and this subset cannot be expressed with MIME type parameters.
      In these cases the attribute 3gpAccept-Subset is used to describe
      support for a subset of a specific content type. If a subset of a
      specific content type is declared in 3gpAccept-Subset, this means that
      3gpAccept-Subset has precedence over 3gpAccept. 3gpAccept shall always
      include the corresponding content types for which 3gpAccept-Subset
      specifies subsets of. No legal values are currently defined.

      Type: Literal (bag)
      Resolution: Locked
    </rdfs:comment>
  </rdf:Description>

<!-- ****************************************************************** -->
<!-- ***** Component: PssSmil ***** -->

  <rdf:Description ID="SmilAccept">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#PssSmil"/>
    <rdfs:comment>
      Description: List of content types (MIME types) that can be part of a
      SMIL presentation. The content types included in this attribute can be
      rendered in a SMIL presentation. If video/3gpp (or audio/3gpp) is
      included, downloaded 3GP files can be included in a SMIL presentation.
      Details on the 3GP file support can then be found in the 3gpFileFormat
      component. If the identifier "Streaming-Media" is included, streaming
      media can be included in the SMIL presentation. Details on the
      streaming support can then be found in the Streaming component.
      For each content type a set of supported parameters can be given.
      A content type that supports multiple parameter sets may occur several
      times in the list. Legal values are lists of MIME types with related
      parameters and the "Streaming-Media" identifier.

      Type: Literal (bag)
      Resolution: Append
      Examples: "image/gif,image/jpeg,Streaming-Media"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="SmilAccept-Subset">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#PssSmil"/>
    <rdfs:comment>
      Description: List of content types for which the PSS application
      supports a subset. MIME types can in most cases effectively be used to
      express variations in support for different media types. Many MIME types
      have several parameters that can be used for this purpose. There may
      exist content types for which the PSS application only supports a subset
      and this subset cannot be expressed with MIME-type parameters. In these
      cases the attribute SmilAccept-Subset is used to describe support for a
      subset of a specific content type. If a subset of a specific content type
      is declared in SmilAccept-Subset, this means that SmilAccept-Subset has
      precedence over SmilAccept. SmilAccept shall always include the
      corresponding content types for which SmilAccept-Subset specifies subsets
      of.

      The following values are defined:
        - "JPEG-PSS": Only the two JPEG modes described in clause 7.5 of the
                      specifictaion are supported.
        - "SVG-Tiny"
        - "SVG-Basic"

      Subset identifiers and corresponding semantics shall only be defined by
      the TSG responsible for the present document.

      Type: Literal (bag)
      Resolution: Append
      Examples: "JPEG-PSS,SVG-Tiny"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="SmilBaseSet">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdfs:domain rdf:resource="#PssSmil"/>
    <rdfs:comment>
      Description: Indicates a base set of SMIL 2.0 modules that the client
      supports. Leagal values are the following pre-defined identifiers:
      "SMIL-3GPP-R4" and "SMIL-3GPP-R5" indicate all SMIL 2.0 modules required
      for scene-description support according to clause 8 of Release 4 and
      Release 5, respectively, of TS 26.234. "SMIL-3GPP-R6" indicates all
      SMIL 2.0 modules required for scene description support according to
      clause 8 of the specification and to Release 6 of TS 26.246 [52].

      Type: Literal
      Resolution: Locked
      Examples: "SMIL-3GPP-R4", "SMIL-3GPP-R5"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="SmilModules">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
    <rdfs:domain rdf:resource="#PssSmil"/>
    <rdfs:comment>
      Description: This attribute defines a list of SMIL 2.0 modules
      supported by the client. If the SmilBaseSet is used those modules
      do not need to be explicitly listed here. In that case only
      additional module support needs to be listed. Legal values are all
      SMIL 2.0 module names defined in the SMIL 2.0 recommendation [31],
      section 2.3.3, table 2.

      Type: Literal (bag)
      Resolution: Locked
      Examples: "BasicTransitions,MulitArcTiming"
    </rdfs:comment>
  </rdf:Description>

</rdf:RDF>


~~<?xml version="1.0"?>~~

~~<!--~~
     ~~This document is the RDF Schema for streaming-specific vocabulary~~
     ~~as defined in 3GPP TS 26.234 Rel.5 (in the following "the~~
     ~~specification").~~

     ~~The URI for unique identification of this RDF Schema is~~
       ~~http://www.3gpp.org/profiles/PSS/ccppschema-PSS5~~

     ~~This RDF Schema includes the same information as the respective~~
     ~~chapter of the specification. Greates care has been taken to keep~~
     ~~the two documents consistence. However, in case of any divergence~~
     ~~the specification takes presidence.~~

     ~~All reference in this RDF Schmea are to be interpreted relative to~~
     ~~the specification. This means all references using the form~~
     ~~[ref] are defined in chapter 2 "References of the~~
     ~~specification. All other references refer to parts within that~~
     ~~document.~~

     ~~Note: This Schemas has been aligned in structure and base~~
     ~~vocabulary to the RDF Schema used by UAProf [40].~~

~~-->~~

~~<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"~~
         ~~xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema" >~~


 ~~<!-- ****************************************************************** -->~~
~~<!-- ***** Properties shared among the components***** -->~~

  ~~<rdf:Description ID="defaults">~~
    ~~<rdfs:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
    ~~<rdfs:domain rdf:resource="Streaming"/>~~
    ~~<rdfs:comment>~~
      ~~An attribute used to identify the default capabilities.~~
    ~~</rdfs:comment>~~
  ~~</rdf:Description>~~


~~<!-- ****************************************************************** -->~~
~~<!-- ***** Component Definitions ***** -->~~

  ~~<rdf:Description ID="Streaming">~~
    ~~<rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>~~
    ~~<rdfs:subClassOf rdf:resource="http://www.wapforum.org/UAPROF/ccppschema-20010330#Component"/>~~
    <rdfs:label>Component: Streaming</rdfs:label>
    <rdfs:comment>
      The Streaming component specifies the base vocabulary for
      PSS. PSS servers supporting capability exchange should
      understand the attributes in this component as explained in
      detail in 3GPP TS 26.234 rel. 5.
    </rdfs:comment>
  </rdf:Description>

<!-- **
     ** In the following property definitions, the defined types
     ** are as follows:
     **
     ** Number: A positive integer
     ** [0-9]+
     ** Boolean: A yes or no value
     ** Yes|No
     ** Literal: An alphanumeric string
     ** [A-Za-z0-9/.\-_]+
     ** Dimension: A pair of numbers
     ** [0-9]+x[0-9]+
     **
-->


<!-- ****************************************************************** -->
<!-- ***** Component: Streaming ***** -->

<rdf:Description ID="AudioChannels">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>
  <rdfs:domain rdf:resource="#Streaming"/>
  <rdfs:comment>
    Description: This attribute describes the stereophonic capability of the natural audio device. 
The only legal values are "Mono" and "Stereo".

    Type: Literal
    Resolution: Locked
    Examples: "Mono", "Stereo"
  </rdfs:comment>
</rdf:Description>


<rdf:Description ID="VideoPreDecoderBufferSize">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>
  <rdfs:domain rdf:resource="#Streaming"/>
  <rdfs:comment>
    Description: This attribute signals if the optional video
    buffering requirements defined in Annex G are supported. It also
    defines the size of the hypothetical pre-decoder buffer defined in
    Annex G. A value equal to zero means that Annex G is not
    supported. A value equal to one means that Annex G is
    supported. In this case the size of the buffer is the default size
    defined in Annex G.  A value equal to or greater than the default
    buffer size defined in Annex G means that Annex G is supported and
    sets the buffer size to the given number of octets. Legal values are all
    integer values equal to or greater than zero. Values greater than
    one but less than the default buffer size defined in Annex G are
    not allowed.

    Type: Number
    Resolution: Locked
    Examples: "0", "4096"
  </rdfs:comment>
</rdf:Description>


<rdf:Description ID="VideoInitialPostDecoderBufferingPeriod">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>
  <rdfs:domain rdf:resource="#Streaming"/>
  <rdfs:comment>
    Description: If Annex G is not supported, the attribute has no
    meaning. If Annex G is supported, this attribute defines the
    maximum initial post-decoder buffering period of video. Values are
    interpreted as clock ticks of a 90-kHz clock. In other words, the
    value is incremented by one for each 1/90 000 seconds. For
    example, the value 9000 corresponds to 1/10 of a second initial
    post-decodder buffering. Legal valaues are all integer value equal
    to or greater than zero.
```

```
~~    Type: Number~~
~~    Resolution: Locked~~
~~    Examples: <VideoInitialPostDecoderBufferingPeriod>~~
~~                9000~~
~~          </VideoInitialPostDecoderBufferingPeriod>~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~

~~<rdf:Description ID=" VideoDecodingByteRate ">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
   ~~Description: If Annex G is not supported, the attribute has no meaning. If Annex G is supported,~~
   ~~this attribute defines the peak decoding byte rate the PSS client is able to support. In other~~
   ~~words, the PSS client fulfils the requirements given in Annex G with the signalled peak decoding~~
   ~~byte rate. The values are given in bytes per second and shall be greater than or equal to 8000.~~
   ~~According to Annex G, 8000 is the default peak decoding byte rate for the mandatory video codec~~
   ~~profile and level (H.263 Profile 0 Level 10).Legal values are integer value greater than or equal~~
   ~~to 8000.~~

~~    Type: Number~~
~~    Resolution: Locked~~
~~    Examples: <VideoDecodingByteRate>16000</VideoDecodingByteRate>~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~

~~<rdf:Description ID=" MaxPolyphony">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
~~    Description: Attribute definition:  The MaxPolyphony attribute refers to the maximal polyphony~~
~~    that the synthetic audio device supports as defined in [44]. Legal values are integer between 5~~
~~    to 24.~~
~~    NOTE:       MaxPolyphony attribute can be used to signal the maximum polyphony capabilities~~
                ~~supported by the PSS client. This is a complementary mechanism for the delivery of~~
                ~~compatible SP-MIDI content and thus the PSS client is required to support Scalable~~
                ~~Polyphony MIDI i.e. Channel Masking defined in [44].~~

~~    Type: Number~~
~~    Resolution: Locked~~
~~    Examples: <MaxPolyphony>8</MaxPolyphony>~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~

~~<rdf:Description ID="PssAccept">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
~~    Description: List of content types (MIME types) the PSS~~
~~    application supports. Both CcppAccept (SoftwarePlatform, UAProf)~~
~~    and PssAccept can be used but if PssAccept is defined it has~~
~~    precedence over CcppAccept and a PSS application shall then use~~
~~    PssAccept.~~

~~    Type: Literal (bag)~~
~~    Resolution: Append~~
~~    Examples: "audio/AMR-WB;octet-alignment,application/smil"~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~


~~<rdf:Description ID="PssAccept-Subset">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
~~    Description: List of content types for which the PSS application~~
~~    supports a subset. MIME-types can in most cases effectively be~~
~~    used to express variations in support for different media~~
~~    types. Many MIME-types, e.g. AMR-NB has several parameters that~~
~~    can be used for this purpose. There may exist content types for~~
~~    which the PSS application only supports a subset and this subset~~
~~    can not be expressed with MIME-type parameters. In these cases the~~
~~    attribute PssAccept-Subset is used to describe support for a~~
~~    subset of a specific content type. If a subset of a specific~~
~~    content type is declared in PssAccept-Subset, this means that~~
~~    PssAccept-Subset has precedence over both PssAccept and CcppAccept.~~
~~    PssAccept and/or CcppAccept shall always include the corresponding~~
~~    content types for which PSSAccept-Subset specifies subsets of.~~
~~    This is to ensure compatibility with those content servers that~~
~~    do not understand the PssAccept-Subset attribute but do understand e.g. CcppAccept.~~

~~    This is illustrated with an example. If PssAccept="audio/AMR",~~
~~    "image/jpeg" and PssAccept-Subset="JPEG-PSS" then "audio/AMR"~~
~~    and JPEG Base line is supported. "image/jpeg" in PssAccept is of no~~
~~    importance since it is related to "JPEG-PSS" in PssAccept-Subset.~~
~~    Subset identifiers and corresponding semantics shall only be defined by~~
~~    the TSG responsible for the present document. The following values are defined:~~
~~    -   "JPEG-PSS": Only the two JPEG modes described in clause 7.5 of the present~~
~~        document are supported.~~
~~    -   "SVG-Tiny"~~
~~    -   "SVG-Basic"~~
~~    Legal values are subset identifiers defined by the specification.~~

~~    Type: Literal (bag)~~
~~    Resolution: Locked~~
~~    Examples: "JPEG-PSS","SVG-Tiny","SVG-Basic"~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~


~~<rdf:Description ID="PssVersion">~~
~~e  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
~~    Description: Latest PSS version supported by the client. Legal~~
~~    values are "3GPP-R4", "3GPP-R5" and so forth.~~

~~    Type: Literal~~
~~    Resolution: Locked~~
~~    Examples: "3GPP-R4", "3GPP-R5"~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~


~~<rdf:Description ID="RenderingScreenSize">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
~~    Description: The rendering size of the device's screen in unit of~~
~~    pixels. The horizontal size is given followed by the vertical~~
~~    size. Legal values are pairs of integer values equal or greater~~
~~    than zero. A value equal "0x0"means that there exist no display or~~
~~    just textual output is supported.~~

~~    Type: Dimension~~
~~    Resolution: Locked~~
~~    Examples: "160x120"~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~


~~<rdf:Description ID="SmilBaseSet">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdfs:domain rdf:resource="#Streaming"/>~~
~~  <rdfs:comment>~~
~~    Description: Indicates a base set of SMIL 2.0 modules that the~~
~~    client supports. Leagal values are the following pre-defined~~
~~    identifiers: "SMIL-3GPP-R4" indicates all SMIL 2.0~~
~~    modules required for scene description support according to clause~~
~~    8 of Release 4 of TS 26.234. "SMIL-3GPP-R5" indicates all SMIL 2.0~~
~~    modules required for scene description support according to clause~~
~~    8 of the specification.~~

~~    Type: Literal~~
~~    Resolution: Locked~~
~~    Examples: "SMIL-3GPP-R4", "SMIL-3GPP-R5"~~
~~  </rdfs:comment>~~
~~</rdf:Description>~~


~~<rdf:Description ID="SmilModules">~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdfschema#Property"/>~~
~~  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>~~
```

```
  ~~<rdfs:domain rdf:resource="#Streaming"/>~~
  ~~<rdfs:comment>~~
    ~~Description: This attribute defines a list of SMIL 2.0 modules~~
    ~~supported by the client. If the SmilBaseSet is used those modules~~
    ~~do not need to be explicitly listed here. In that case only~~
    ~~additional module support needs to be listed. Legal values are all~~
    ~~SMIL 2.0 module names defined in the SMIL 2.0 recommendation [31],~~
    ~~section 2.3.3, table 2.~~

    ~~Type: Literal (bag)~~
    ~~Resolution: Locked~~
    ~~Examples: "BasicTransitions,MulitArcTiming"~~
  ~~</rdfs:comment>~~
~~</rdf:Description>~~

~~</rdf:RDF>~~
```

Annex G (normative): Buffering of video

G.1 Introduction

This annex describes video buffering requirements in the PSS. As defined in clause 7.4 of the present document, support for the annex is optional and may be signalled in the PSS capability exchange and in the SDP. This is described in clause 5.2 and clause 5.3.3 of the present document. When the annex is in use, the content of the annex is normative. In other words, PSS clients shall be capable of receiving an RTP packet stream that complies with the specified buffering model and PSS servers shall verify that the transmitted RTP packet stream complies with the specified buffering model.

G.2 PSS Buffering Parameters

The behaviour of the PSS buffering model is controlled with the following parameters: the initial pre-decoder buffering period, the initial post-decoder buffering period, the size of the hypothetical pre-decoder buffer, the peak decoding byte rate, and the decoding macroblock rate. The default values of the parameters are defined below.

- The default initial pre-decoder buffering period is 1 second.
- The default initial post-decoder buffering period is zero.
- The default size of the hypothetical pre-decoder buffer is defined according to the maximum video bit-rate according to the table below:

Table G.1: Default size of the hypothetical pre-decoder buffer

Maximum video bit-rate	Default size of the hypothetical pre-decoder buffer
65536 bits per second	20480 bytes
131072 bits per second	40960 bytes
Undefined	51200 bytes

- The maximum video bit-rate can be signalled in the media-level bandwidth attribute of SDP as defined in clause 5.3.3 of this document. If the video-level bandwidth attribute was not present in the presentation description, the maximum video bit-rate is defined according to the video coding profile and level in use.
- The size of the hypothetical post-decoder buffer is an implementation-specific issue. The buffer size can be estimated from the maximum output data rate of the decoders in use and from the initial post-decoder buffering period.
- By default, the peak decoding byte rate is defined according to the video coding profile and level in use. For example, H.263 Level 10 requires support for bit-rates up to 64000 bits per second. Thus, the peak decoding byte rate equals to 8000 bytes per second.
- The default decoding macroblock rate is defined according to the video coding profile and level in use. If MPEG-4 Visual is in use, the default macroblock rate equals to VCV decoder rate. If H.263 is in use, the default macroblock rate equals to (1 / minimum picture interval) multiplied by number of macroblocks in maximum picture format. For example, H.263 Level 10 requires support for picture formats up to QCIF and minimum picture interval down to 2002 / 30000 sec. Thus, the default macroblock rate would be 30000 x 99 / 2002 ≈ 1484 macroblocks per second.

PSS clients may signal their capability of providing larger buffers and faster peak decoding byte rates in the capability exchange process described in clause 5.2 of the present document. The average coded video bit-rate should be smaller than or equal to the bit-rate indicated by the video coding profile and level in use, even if a faster peak decoding byte rate were signalled.

Initial parameter values for each stream can be signalled within the SDP description of the stream. Signalled parameter values override the corresponding default parameter values. The values signalled within the SDP description guarantee pauseless playback from the beginning of the stream until the end of the stream (assuming a constant-delay reliable transmission channel).

PSS servers may update parameter values in the response for an RTSP PLAY request. If an updated parameter value is present, it shall replace the value signalled in the SDP description or the default parameter value in the operation of the PSS buffering model. An updated parameter value is valid only in the indicated playback range, and it has no effect after that. Assuming a constant-delay reliable transmission channel, the updated parameter values guarantee pauseless playback of the actual range indicated in the response for the PLAY request. The indicated pre-decoder buffer size and initial post-decoder buffering period shall be smaller than or equal to the corresponding values in the SDP description or the corresponding default values, whichever ones are valid. The following header fields are defined for RTSP:

- x-predecbufsize:<size of the hypothetical pre-decoder buffer>
  This gives the suggested size of the Annex G hypothetical pre-decoder buffer in bytes.

- x-initpredecbufperiod:<initial pre-decoder buffering period>
  This gives the required initial pre-decoder buffering period specified according to Annex G. Values are interpreted as clock ticks of a 90-kHz clock. That is, the value is incremented by one for each 1/90 000 seconds. For example, value 180 000 corresponds to a two second initial pre-decoder buffering.

- x-initpostdecbufperiod:<initial post-decoder buffering period>
  This gives the required initial post-decoder buffering period specified according to Annex G. Values are interpreted as clock ticks of a 90-kHz clock.

These header fields are defined for the response of an RTSP PLAY request only. Their use is optional.

The following example plays the whole presentation starting at SMPTE time code 0:10:20 until the end of the clip. The playback is to start at 15:36 on 23 Jan 1997. The suggested initial ~~post~~pre-decoder buffering period is half a second.

```
C->S: PLAY rtsp://audio.example.com/twister.en RTSP/1.0
      CSeq: 833
      Session: 12345678
      Range: smpte=0:10:20-;time=19970123T153600Z
      User-Agent: TheStreamClient/1.1b2

S->C: RTSP/1.0 200 OK
      CSeq: 833
      Date: 23 Jan 1997 15:35:06 GMT
      Range: smpte=0:10:22-;time=19970123T153600Z
      x-initpredecbufperiod: 45000
```

G.3 PSS server buffering verifier

The PSS server buffering verifier is specified according to the PSS buffering model. The model is based on two buffers and two timers. The buffers are called the hypothetical pre-decoder buffer and the hypothetical post-decoder buffer. The timers are named the decoding timer and the playback timer.

The PSS buffering model is presented below.

1. The buffers are initially empty.

2. A PSS Server adds each transmitted RTP packet having video payload to the pre-decoder buffer immediately when it is transmitted. All protocol headers at RTP or any lower layer are removed.

3. Data is not removed from the pre-decoder buffer during a period called the initial pre-decoder buffering period. The period starts when the first RTP packet is added to the buffer.

4. When the initial pre-decoder buffering period has expired, the decoding timer is started from a position indicated in the previous RTSP PLAY request.

5. Removal of a video frame is started when both of the following two conditions are met: First, the decoding timer has reached the scheduled playback time of the frame. Second, the previous video frame has been totally removed from the pre-decoder buffer.

6. The duration of frame removal is the larger one of the two candidates: The first candidate is equal to the number of macroblocks in the frame divided by the decoding macroblock rate. The second candidate is equal to the number of bytes in the frame divided by the peak decoding byte rate. When the coded video frame has been removed from the pre-decoder buffer entirely, the corresponding uncompressed video frame is located into the post-decoder buffer.

7. Data is not removed from the post-decoder buffer during a period called the initial post-decoder buffering period. The period starts when the first frame has been placed into the post-decoder buffer.

8. When the initial post-decoder buffering period has expired, the playback timer is started from the position indicated in the previous RTSP PLAY request.

9. A frame is removed from the post-decoder buffer immediately when the playback timer reaches the scheduled playback time of the frame.

10. Each RTSP PLAY request resets the PSS buffering model to its initial state.

A PSS server shall verify that a transmitted RTP packet stream complies with the following requirements:

- The PSS buffering model shall be used with the default or signalled buffering parameter values. Signalled parameter values override the corresponding default parameter values.

- The occupancy of the hypothetical pre-decoder buffer shall not exceed the default or signalled buffer size.

- Each frame shall be inserted into the hypothetical post-decoder buffer before or on its scheduled playback time.

G.4 PSS client buffering requirements

When the annex is in use, the PSS client shall be capable of receiving an RTP packet stream that complies with the PSS server buffering verifier, when the RTP packet stream is carried over a constant-delay reliable transmission channel. Furthermore, the video decoder of the PSS client, which may include handling of post-decoder buffering, shall output frames at the correct rate defined by the RTP time-stamps of the received packet stream.

Annex H (informative): Content creator guidelines for the synthetic audio medium type

It is recommended that the first element of the MIP (Maximum Instantaneous Polyphony) message of the SP-MIDI content intended for synthetic audio PSS/MMS should be no more than 5. For instance the following MIP figures {4, 9, 10, 12, 12, 16, 17, 20, 26, 26, 26} complies with the recommendation whereas {6, 9, 10, 12, 12, 16, 17, 20, 26, 26, 26} does not.

Annex I (informative): (void) ~~SP MIDI Device 5–24 Note Profile for 3GPP, SP-MIDI implementation guideline using a non-compliant hardware~~

~~I.1 Introduction~~

~~This informative annex describes some implementation guidelines intended for SP-MIDI device 5-24 Note Profile for 3GPP [45]. These guidelines are here to give the possibility for manufacturers to develop early SP-MIDI implementations using MIDI hardware available at the time of the approval of release 5. These guidelines are valid only for release 5 implementations of SP-MIDI and are expected to be removed . It should be noted that these guidelines may reduce the musical performance of the synthesiser depending on the content and should be used with extreme caution.~~

~~I.2 Guidelines~~

~~I.2.1 Support of multiple rhythm channels~~

~~Scalable Polyphony synthesisers conformant to this Profile shall support at least two MIDI Channels that can function as Rhythm Channels, to enable a fluent scalable polyphony implementation.~~

~~If the two rhythm Channels are not natively supported by the MIDI hardware, the SP-MIDI player could redirect the events intended to the additional rhythm channels toward the default rhythm channel (MIDI channel 10). The rendering of the SP-MIDI content should not be affected until different Channel settings (e.g. Channel Volume, Bank Setting, Panning etc.) are applied to the different rhythm Channels. It is recommended that only Channel settings intended for the default rhythm channel be applied.~~

~~I.2.2 Support of individual stereophonic panning~~

~~When the support of individual stereophonic panning is not possible by the stereophonic MIDI synthesiser, central panning should be used as default instead.~~

Annex J (informative): Mapping of SDP parameters to UMTS QoS parameters

This Annex gives recommendation for the mapping rules needed by the PSS applications to request the appropriate QoS from the UMTS network (see Table J.1).

Table J.1: Mapping of SDP parameters to UMTS QoS parameters for PSS

QoS parameter	Parameter value	comment
Delivery of erroneous SDUs	”No”	
Delivery order	”No”	
Traffic class	"Streaming class"	
Maximum SDU size	1400 bytes	According to RFC 2460 the SDU size must not exceed 1500 octets. A packet size of 1400 guarantees efficient transportation.
Guaranteed bit rate for downlink	1.025 * session bandwidth	This session bandwidth is calculated from the SDP media level bandwidth values.
Maximum bit rate for downlink	Equal or higher to guaranteed bit rate in downlink	
Guaranteed bit rate for uplink	0.025 * session bandwidth	
Maximum bit rate for uplink	Equal or higher to guaranteed bit rate in uplink	
Residual BER	1*10-5	16 bit CRC should be enough
SDU error ratio	1*10-4 or better	
Traffic handling priority	Subscribed traffic handling priority	Ignored
Transfer delay	2 sec.	

Annex X~~K~~ (informative): Change history

Change history							
Date	**TSG SA #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
03-2001	11	SP-010094			Version for Release 4		4.0.0
09-2001	13	SP-010457	001	1	3GPP PSS4 SMIL Language Profile	4.0.0	4.1.0
09-2001	13	SP-010457	002		Clarification of H.263 baseline settings	4.0.0	4.1.0
09-2001	13	SP-010457	003	2	Updates to references	4.0.0	4.1.0
09-2001	13	SP-010457	004	1	Corrections to Annex A	4.0.0	4.1.0
09-2001	13	SP-010457	005	1	Clarifications to chapter 7	4.0.0	4.1.0
09-2001	13	SP-010457	006	1	Clarification of the use of XHTML Basic	4.0.0	4.1.0
12-2001	14	SP-010703	007		Correction of SDP Usage	4.1.0	4.2.0
12-2001	14	SP-010703	008	1	Implementation guidelines for RTSP and RTP	4.1.0	4.2.0
12-2001	14	SP-010703	009		Correction to media type decoder support in the PSS client	4.1.0	4.2.0
12-2001	14	SP-010703	010		Amendments to file format support for 26.234 release 4	4.1.0	4.2.0
03-2002	15	SP-020087	011		Specification of missing limit for number of AMR Frames per Sample	4.2.0	4.3.0
03-2002	15	SP-020087	013	2	Removing of the reference to TS 26.235	4.2.0	4.3.0
03-2002	15	SP-020087	014		Correction to the reference for the XHTML MIME media type	4.2.0	4.3.0
03-2002	15	SP-020087	015	1	Correction to MPEG-4 references	4.2.0	4.3.0
03-2002	15	SP-020087	018	1	Correction to the width field of H263SampleEntry Atom in Section D.6	4.2.0	4.3.0
03-2002	15	SP-020087	019		Correction to the definition of "b=AS"	4.2.0	4.3.0
03-2002	15	SP-020087	020		Clarification of the index number's range in the referred MP4 file format	4.2.0	4.3.0
03-2002	15	SP-020087	021		Correction of SDP attribute 'C='	4.2.0	4.3.0
03-2002	15	SP-020173	023		References to "3GPP AMR-WB codec" replaced by "ITU-T Rec. G.722.2" and "RFC 3267"	4.2.0	4.3.0
03-2002	15	SP-020088	022	2	Addition of Release 5 functionality	4.3.0	5.0.0
06-2002	16	SP-020226	024	1	Correction to Timed Text	5.0.0	5.1.0
06-2002	16	SP-020226	026	3	Mime media type update	5.0.0	5.1.0
06-2002	16	SP-020226	027		Corrections to the description of Sample Description atom and Timed Text Format	5.0.0	5.1.0
06-2002	16	SP-020226	029	1	Corrections Based on Interoperability Issues	5.0.0	5.1.0
09-2002	17	SP-020439	030	2	Correction regarding support for Timed Text	5.1.0	5.2.0
09-2002	17	SP-020439	032	3	Required RTSP header support	5.1.0	5.2.0
09-2002	17	SP-020439	034	1	Including bitrate information for H.263	5.1.0	5.2.0
09-2002	17	SP-020439	035	1	RTCP Reports and Link Aliveness in Ready State	5.1.0	5.2.0
09-2002	17	SP-020439	036	2	Correction on media and session-level bandwidth fields in SDP	5.1.0	5.2.0
09-2002	17	SP-020439	037	2	Correction on usage of MIME parameters for AMR	5.1.0	5.2.0
09-2002	17	SP-020439	038	1	Correction of Mapping of SDP parameters to UMTS QoS parameters (Annex J)	5.1.0	5.2.0
12-2002	18	SP-020694	039	2	Addition regarding IPv6 support in SDP	5.2.0	5.3.0
12-2002	18	SP-020694	040		Code points for H.263	5.2.0	5.3.0
12-2002	18	SP-020694	041	2	File format 3GP based on ISO and not MP4	5.2.0	5.3.0
12-2002	18	SP-020694	044	1	SMIL authoring instructions	5.2.0	5.3.0
12-2002	18	SP-020694	045	1	Client usage of bandwidth parameter at the media level in SDP	5.2.0	5.3.0
12-2002	18	SP-020694	047	1	SMIL Language Profile	5.2.0	5.3.0
12-2002	18	SP-020694	050	1	Usage of Multiple Media Sample Entries in Media Tracks of 3GP files	5.2.0	5.3.0
12-2002	18	SP-020694	051	1	Progressive download of 3GP files	5.2.0	5.3.0
03-2003	19	SP-030091	052	1	SDP bandwidth modifier for RTCP bandwidth	5.3.0	5.4.0
03-2003	19	SP-030091	053		Specification of stream control URLs in SDP files	5.3.0	5.4.0

03-2003	19	SP-030091	054		Clarification of multiple modifiers for timed text	5.3.0	5.4.0
03-2003	19	SP-030091	056	4	Correction of wrong references	5.3.0	5.4.0
03-2003	19	SP-030091	057	2	Correction of signalling frame size for H.263 in SDP	5.3.0	5.4.0
06-2003	20	SP-030217	058	1	SMIL supported event types	5.4.0	5.5.0
06-2003	20	SP-030217	060		Correction to the Content Model of the SMIL Language Profile	5.4.0	5.5.0
09-2003	21	SP-030448	061	1	Correction on session bandwidth for RS and RR RTCP modifiers	5.5.0	5.6.0
09-2003	21	SP-030448	062	1	Correction of ambiguous range headers in SDP	5.5.0	5.6.0
09-2003	21	SP-030448	063	1	Timed-Text layout example	5.5.0	5.6.0
09-2003	21	SP-030448	064		Correction of ambiguity in RTP timestamps handling after PAUSE/PLAY RTSP requests	5.5.0	5.6.0
09-2003	21	SP-030448	065		Correction of obsolete RTP references	5.5.0	5.6.0
09-2003	21	SP-030448	066	1	Correction of wrong reference	5.5.0	5.6.0
09-2003	21	SP-030448	067		Missing signaling of live content	5.5.0	5.6.0