Technical Specification Group Services and System Aspects **TSGS#23(04)0652**
Meeting #25, Palm Springs, USA
13-16 September 2004

**Source: TSG-SA WG4**

**Title: CR TS 26.234 on Additional Release-6 updates to PSS Protocols and codecs (Release 6)**

## Document for: Approval

## Agenda Item: 7.4.3

The following CR, agreed at the TSG-SA WG4 meeting #32, is presented to TSG SA #25 for approval.

| Spec | CR | Rev | Phase | Subject | Cat | Vers | WG | Meeting | S4 doc |
|---|---|---|---|---|---|---|---|---|---|
| 26.234 | 070 | 1 | Rel-6 | Additional Release-6 updates to PSS Protocols and codecs | B | 6.0.0 | S4 | TSG-SA WG4#32 | S4-040506 |

**3GPP TSG-SA4 Meeting #32** ***Tdoc ⌘S4-040506***
**Prague, Czech Rebublic, 16-20 August 2004**

*CR-Form-v7*

# CHANGE REQUEST

⌘ **26.234** **CR** **070** ⌘**rev** **1** ⌘ Current version: **6.0.0** ⌘

*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:*** | ⌘ | Additional Release-6 updates to PSS Protocols and Codecs |
| ***Source:*** | ⌘ | TSG SA WG4 |
| ***Work item code:*** | ⌘ | PSSrel6-Stage3 |
| ***Date:*** | ⌘ | 14/09/2004 |
| ***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:*** | ⌘ | Update of Release-6 functionality |
| ***Summary of change:*** | ⌘ | The following updates have been implemented:<br>• DRM (confidentiality and integrity protection) included<br>• RTP retransmisson included<br>• RTP transport of timed text included<br>• PSS UAProf vocabulary updated<br>• MIME type references updated<br>• Support for 128 kbps video included<br>• Client buffer feedback mechanism updated<br>• Quality of Experience metrics added<br>• Editoral changes |
| ***Consequences if not approved:*** | ⌘ | Release-6 will not contain the new features added above. |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 2, 3.2, 4, 5.2.3.2, 5.2.3.2.2, 5.2.3.2.3, 5.2.3.2.4, 5.2.3.3, 5.3.1, 5.3.2.2, 5.3.2.3.2, 5.3.3.2, 5.3.3.5, 5.4, 6.2.1, 6.2.2, 6.2.3, 6.2.3.2, 6.2.3.3, 6.2.3.3.x, 6.2.4, 7.4, 7.7, 7.9, 7.10, 10.2.3, 10.3, 11.2, 11.2.5, 11.2.6, A.1, A.3.3, A.4.3, A.4.7, G.2, F, K |

| ***Other specs affected:*** | ⌘ | Y | N | | ⌘ | |
|---|---|---|---|---|---|---|
| | | X | | Other core specifications | | CR 26.244 003 |
| | | | X | Test specifications | | |
| | | | X | O&M Specifications | | |

| | | |
|---|---|---|
| ***Other comments:*** | ⌘ | |

# 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] (void)~~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 (~~19~~02/98): "Video coding for low bit rate communication".

[23] ITU-T Recommendation H.263 – Annex X (~~2001~~03/04): "Annex X: Profiles and levels definition".

[24] ISO/IEC 14496-2:2004~~1~~: "Information technology – Coding of audio-visual objects – Part 2: Visual".

[25] (void)~~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)

[36] (void)

[37] (void)

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

[39] 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 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)

[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)".

[51] 3GPP TS 26.245: "Transparent end-to-end packet switched streaming service (PSS); Timed text format".

[52] 3GPP TS 26.246: "Transparent end-to-end packet switched streaming service (PSS); 3GPP SMIL Language Profile".

[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-11~~08~~.txt, ~~January~~August 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.

[72] IETF RFC 3711: "The Secure Real-time Transport Protocol (SRTP)", Baugher M. et al, March 2004.

[73] Bellovin, S., "Problem Areas for the IP Security Protocols" in Proceedings of the Sixth Usenix Unix Security Symposium, pp. 1-16, San Jose, CA, July 1996

[74] Open Mobile Alliance: "DRM Specification 2.0".

[75] Open Mobile Alliance: "DRM Content Format V 2.0".

[76] IETF RFC 3675: "IPv6 Jumbograms", Borman D., Deering S. and Hinden R., August 1999.

[77] NIST, "Advanced Encryption Standard (AES)", FIPS PUB 197, http://www.nist.gov/aes/.

[78] IETF RFC 3394: "Advanced Encryption Standard (AES) Key Wrap Algorithm", Schaad J. and Housley R., September 2002.

[79] IETF RFC 3839: "MIME Type Registrations for 3rd Generation Partnership Project (3GPP) Multimedia files", Castagno R. and Singer D., July 2004.

[80] IETF Internet Draft: "RTP Payload Format for 3GPP Timed Text", Rey J. and Matsui Y., http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-3gpp-timed-text-04.txt, July 2004.

[81] IETF Internet Draft: "RTP Retransmission Payload Format", Rey J. et al, http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-retransmission-10.txt, January 2004.

## 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 |
| ADU | Application Data Unit |
| CC/PP | Composite Capability / Preference Profiles |
| DCT | Discrete Cosine Transform |
| DLS | Downloadable Sounds |
| DRM | Digital Rights Management |
| 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 |
| 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 |
| SRTP | The Secure Real-time Transport Protocol |
| 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) |
| W3C | WWW Consortium |
| WML | Wireless Markup Language |
| XHTML | eXtensible Hyper Text Markup Language |
| XMF | eXtensible Music Format |
| XML | eXtensible Markup Language |

# 4 System description

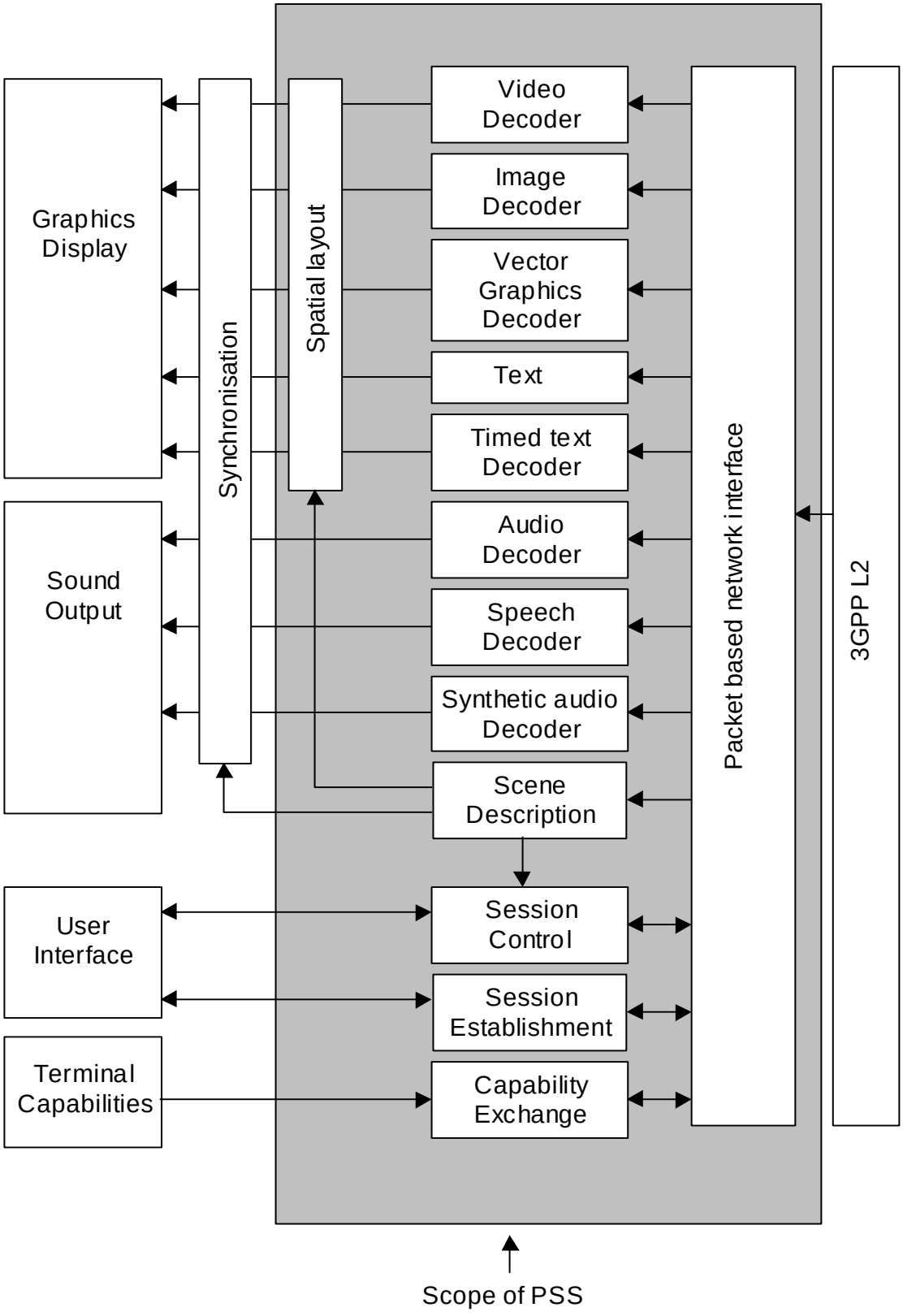



**Figure 1: Functional components of a PSS client**

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.

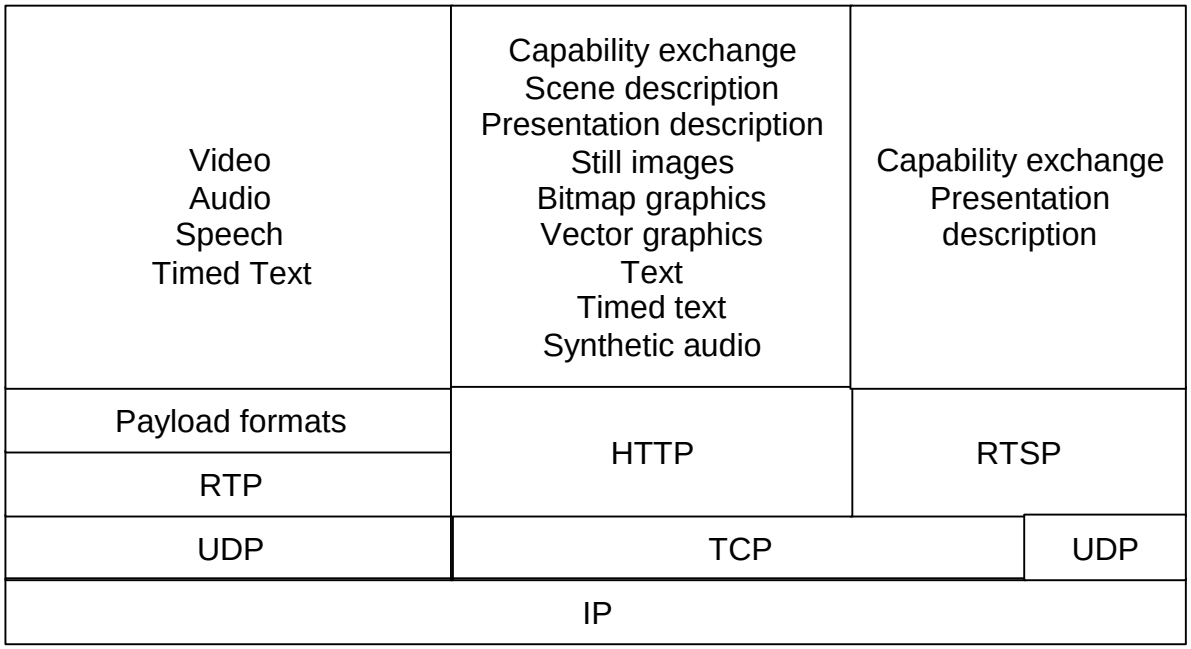
The control related elements are session establishment, capability exchange and session control (see clause 5).

- 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.



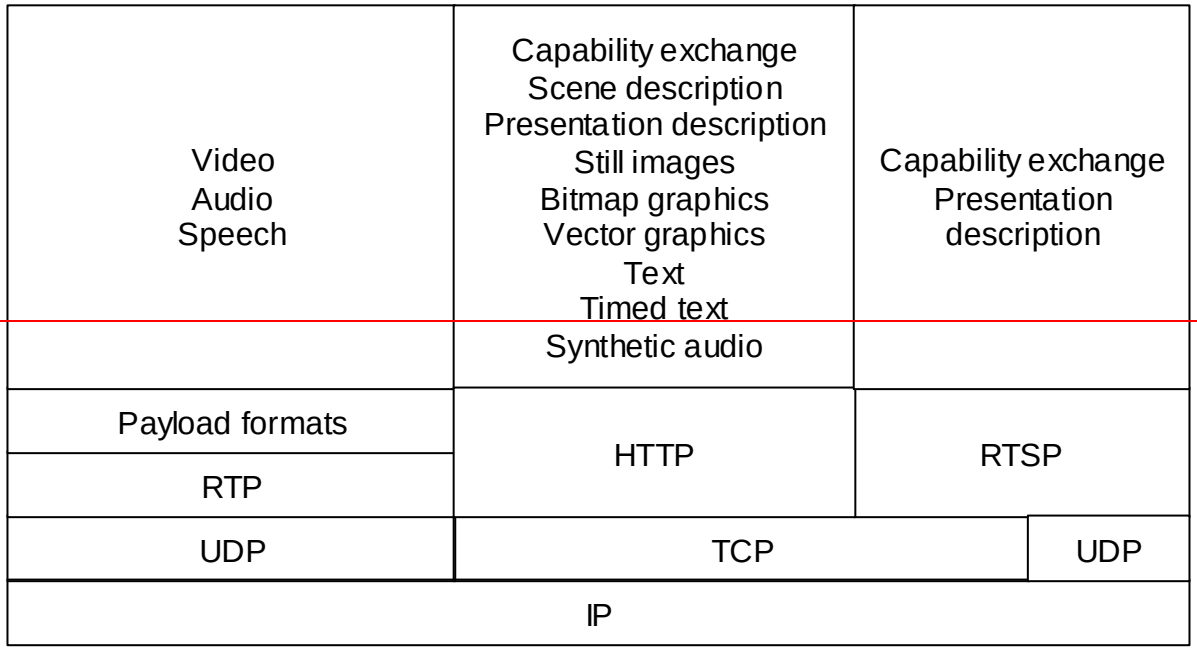




**Figure 2: Overview of the protocol stack**

### 5.2.3.2 PSS base vocabulary

The PSS base vocabulary contains four components called "PssCommon", "Streaming", "~~3gp~~ThreeGPFileFormat" 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 ~~3gp~~ThreeGPFileFormat 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 schema for the PSS base vocabulary can be found in Annex F. Together 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.

The PSS specific components contain a number of attributes expressing capabilities. The following subclauses list all attributes for each component.

#### 5.2.3.2.2 Streaming component

Attribute name: **StreamingAccept**

Attribute definition: 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.

Component: Streaming

Type: Literal (Bag)

Legal values: List of MIME types with related parameters.

Resolution rule: Append

EXAMPLE 1:
```
<StreamingAccept>
  <rdf:Bag>
    <rdf:li>audio/AMR-WB; octet-alignment=1</rdf:li>
    <rdf:li>video/H263-2000; profile=0; level=45~~10~~</rdf:li>
  </rdf:Bag>
</StreamingAccept>
```

Attribute name: **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 have~~s~~ 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.

Component: Streaming

Type: Literal (Bag)

Legal values: No subsets defined.

Resolution rule: Append

Attribute name: **~~3gpp~~ThreeGPPLinkChar**

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: `<~~3gpp~~ThreeGPPLinkChar>Yes</~~3gpp~~ThreeGPPLinkChar>`

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: **RtpRetransmission**

Attribute definition: Indicates whether the device supports RTP retransmission according to clause 6.2.3.3.

Component: Streaming

Type: Literal

Legal values: "Yes", "No"

Resolution rule: Locked

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

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>
</RtpProfiles>
```

Attribute name: **StreamingOmaDrm**

Attribute definition: Indicates whether the device supports streamed OMA DRM protected content as defined by OMA and Annex K.

Component: Streaming

Type: Literal (Bag)

Legal values: OMA Version numbers supported as a floating number. 0.0 indicates no support.

Resolution rule: Locked

EXAMPLE 7:
```
<StreamingOmaDrm>
  <rdf:Bag>
    <rdf:li>2.0</rdf:li>
  </rdf:Bag>
</StreamingOmaDrm>
```

Attribute name: **PSSIntegrity**

Attribute definition: Indicates whether the device supports integrity protection for streamed content as defined by Annex K.2.

Component: Streaming

Type: Literal

Legal values: "Yes", "No"

Resolution rule: Locked

EXAMPLE 8: `<PSSIntegrity>Yes</PSSIntegrity>`

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 168000. According to Annex G, 816000 is the default peak decoding byte rate for the mandatory video codec profile and level (H.263 Profile 0 Level 4510).

Component: Streaming

Type: Number

Legal values: Integer value greater than or equal to 816000.

Resolution rule: Locked

EXAMPLE 7: `<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:
```
<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:
```
<VideoPreDecoderBufferSize>30720</VideoPreDecoderBufferSize>
```

### 5.2.3.2.3 ~~3gp~~ThreeGPFileFormat component

Attribute name: **Brands**

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

Component: ~~3gp~~ThreeGPFileFormat

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: **~~3gp~~ThreeGPAccept**

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: ~~3gp~~ThreeGPFileFormat

Type: Literal (Bag)

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

Resolution rule: Append

EXAMPLE 2:
```
<3gpThreeGPAccept>
  <rdf:Bag>
    <rdf:li>video/H263-2000; profile=0; level=4510</rdf:li>
    <rdf:li>audio/AMR</rdf:li>
    <rdf:li>Timed-Text</rdf:li>
  </rdf:Bag>
</3gpThreeGPAccept>
```

Attribute name: **3gpThreeGPAccept-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 3gpThreeGPAccept-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 3gpThreeGPAccept-Subset, this means that 3gpThreeGPAccept-Subset has precedence over 3gpThreeGPAccept. 3gpThreeGPAccept shall always include the corresponding content types for which 3gpThreeGPAccept-Subset specifies subsets of.

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

Component: 3gpThreeGPFileFormat

Type: Literal (Bag)

Legal values: No subsets defined.

Resolution rule: Append

Attribute name: **ThreeGPOmaDrm**

Attribute definition: List of the OMA DRM versions that is supported to be used for DRM protection of content present in the 3GP file format.

Component: ThreeGPFileFormat

Type: Literal (Bag)

Legal values: OMA DRM version numbers as floating point values. 0.0 indicates no support.

Resolution rule: Locked

EXAMPLE 3:
```
<3gpOMADRM>
  <rdf:Bag>
    <rdf:li>2.0 </rdf:li>
  </rdf:Bag>
</3gpOMADRM>
```

### 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 3gpThreeGPFileFormat 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>Model B~~Lexus~~</Model>
```

Attribute name: **Vendor**

Component: HardwarePlatform

Attribute description: Name of the vendor manufacturing the terminal device

EXAMPLE 6:
```
<Vendor>TerminalManufacturer A~~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>
```

## 5.3.1 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 and timed text belong~~s~~ to the 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.

### 5.3.2.2 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, and, if used, de-interleaving buffer(s) that have~~s~~ this given amount of space for complete application data units (ADU), including the following RTP ~~packets including the RTP~~ header and RTP payload header fields: RTP timestamp, and sequence numbers or decoding order numbers. 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.

#### 5.3.2.3.2 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.

### 5.3.3.2 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, 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, 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.

  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, 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.

  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.

If integrity protection is supported, the following SDP attributes shall be supported by the client and server:

- "a=3GPP-Integrity-Key" according to annex K;
- "a=3GPP-SRTP-Config" according to Annex K;
- "a=3GPP-SDP-Auth" according to Annex K.

If RTP retransmission is supported, the following SDP attribute shall be supported by the client and server:

- "a=rtcp-fb" according to clause 4.2 in [57].

### 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~~ NADU APP packet for reporting ~~of the oldest buffered sequence number~~ the next unit to be decoded, 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 NADU~~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~~ NADU APP packet in at least every 3rd RTCP packet.

## 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 clause 4.2.7 of [62];~~.~~
- 3GPP timed text format [51] MIME media type as defined in clause 7.1 of [80];
- OMA DRM protected streaming media MIME media type as defined in clause K.1.4 in Annex K;
- RTP retransmission payload format MIME media types as defined in clause 8 of [81].

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 [79~~50~~].

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

## 6.2.1 General

The IETF RTP [9] 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.

RTP/UDP/IP transport of ~~continuous media (~~speech, audio and video~~)~~ shall be supported. RTP/UDP/IP transport of timed text should 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 shall support the generic NACK message specified in section 6.2.1 of [57] if RTP retransmission is supported. A PSS client or server is not required to support the other 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.

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

- The Secure Real-time Transport Protocol (SRTP) [72], also called RTP/SAVP.

## 6.2.3 RTP and 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~~ NADU APP packet)

To report the next application data unit to be decoded~~oldest buffered sequence number (OBSN)~~ for ~~bit-rate adaptation~~ buffer status reporting and 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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

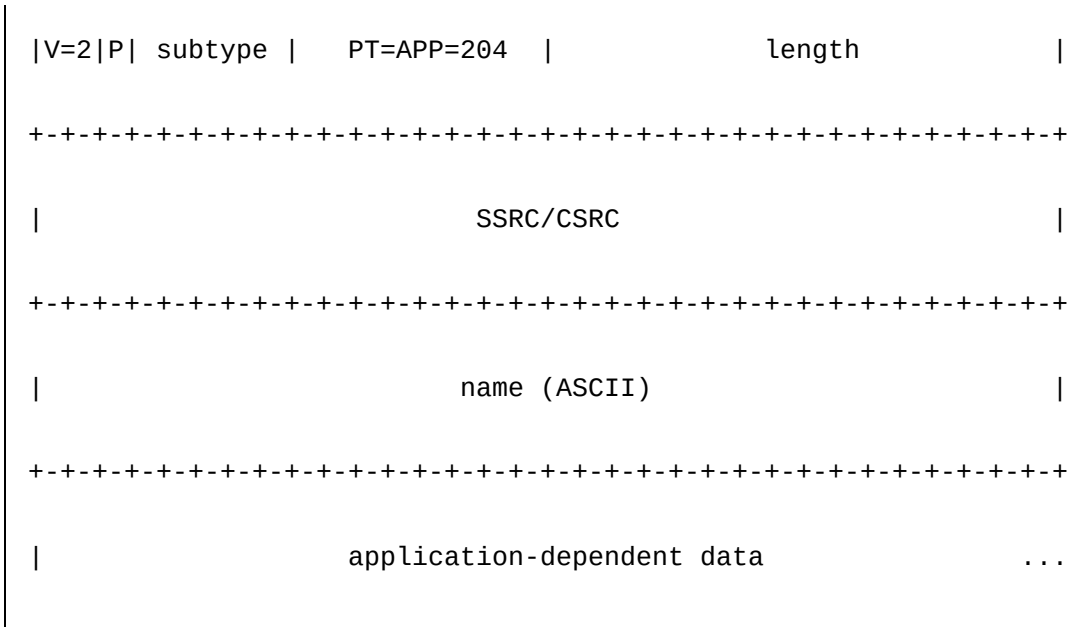



Figure 3: Generic Format of an RTCP APP packet.

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

*name*: The ~~OBSN~~ NADU 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~~ NADU format.
*length*: The number of 32 bit words –1, as defined in RFC 3550 [9]. This means that the field will be 2+~~2~~3*N, where N is the number of sources reported on. The length field will typically be 5~~4~~, i.e. 24~~0~~ 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~~ are a report block, part of either a Receiver Report or a Sender Report, included in the RTCP compound packet. An ~~OBSN~~ NADU APP packet shall not contain any other data format than the one described in figure 4 below.

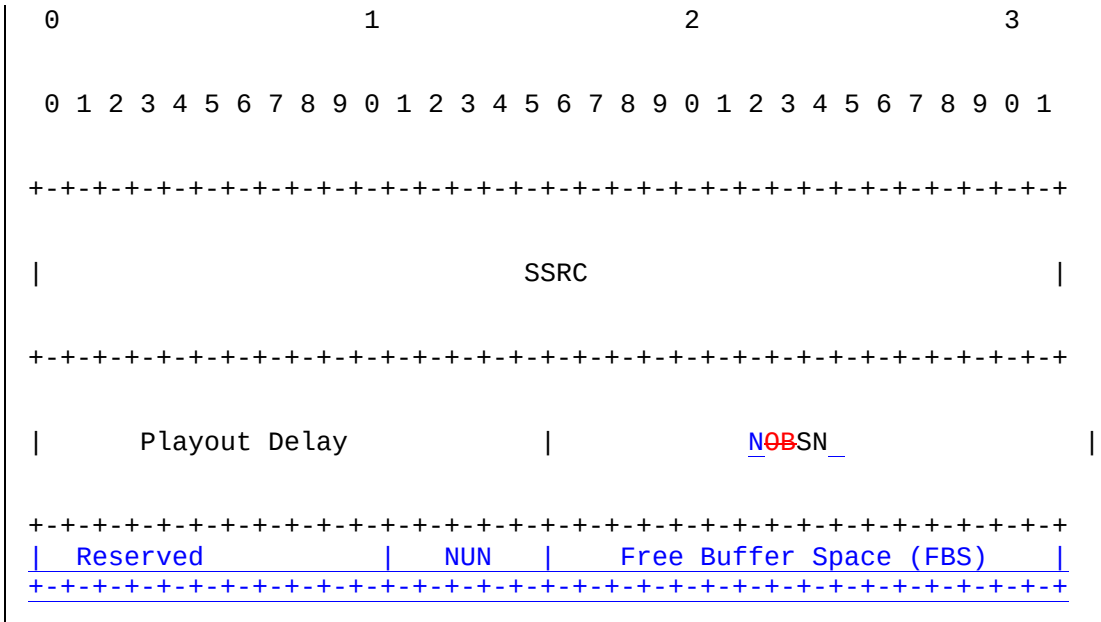



**Figure 4: Data format block for NADU~~OBSN~~ reporting**

*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 (16 bits)*: The difference between the scheduled playout time of the ~~oldest packet~~next ADU to be decoded and the time of sending the ~~OBSN~~ NADU APP packet, as measured by the media playout clock, expressed 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).

*NSN (16 bits)*: The RTP sequence number of the next ADU to be decoded for the SSRC reported on. In the case where the buffer does not contain any packets for this SSRC, the next not yet received sequence number shall be reported, i.e. an NSN 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".

*NUN (5 bits)*: The unit number (within the RTP packet) of the next ADU to be decoded. The first unit in a packet has a unit number equal to zero. The unit number is incremented by one for each ADU in an RTP packet. In the case of an audio codec, an ADU is defined as an audio frame. In the case of H.264 (AVC), an ADU is defined as a NAL unit. In the case of H.263 and MPEG4 Visual Simple Profile, each packet carries a single ADU and the NUN field shall be thus set to zero. Future additions of media encoding or transports capable of having more than one ADU in each RTP payload shall define what shall be counted as an ADU for this format.

*FBS (16 bit)*: The amount of free buffer space available in the client at the time of reporting. The reported free buffer space shall all be part of the buffer space that has been reported as available for adaptation by the 3GPP-Adaptation RTSP header, see clause 5.3.2.2. The amount of free buffer space are reported in number of complete 64 byte blocks, thus allowing for up to 4194304 bytes to be reported as free. If more is available, it shall be reported as the maximal amount available, i.e. 4194304 with a field value 0xffff.

*Reserved (11 bits)*: These bits are not used and shall be set to 0 and shall be ignored by the receiver.

### 6.2.3.3 RTP retransmission

#### 6.2.3.3.1 General

A PSS client should implement RTP retransmission. A PSS client or server implementing RTP retransmission shall implement the payload format, SDP signalling and mechanisms of the RTP retransmission payload format [81]. In addition to the specifications and recommendations in [81], a PSS client and server supporting RTP retransmission shall follow the definitions in the following clauses.

#### 6.2.3.3.2 Multiplexing scheme

The RTP retransmission payload format [81] provides two different schemes for multiplexing the original and the retransmission stream, i.e. session-multiplexing and SSRC-multiplexing. PSS servers shall use SSRC-multiplexing and shall not use session-multiplexing.

#### 6.2.3.3.3 RTCP retransmission request

PSS clients shall use the NACK feedback message format defined in the "Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)" [57] for requesting the retransmission of RTP packets.

Before requesting the retransmission of RTP packets the client should assess whether a requested packet can be decoded in time by checking the latest receiver buffer status. If the client sends RTCP APP packets for client buffer feedback, as defined in section 6.2.3.2, the same assessment should be performed by the server, according to the latest RTCP APP packet it has received.

#### 6.2.3.3.4 Congestion control and usage with rate adaptation

To avoid network congestion due to the additional bandwidth required for the retransmission of lost packets, the available link rate shall be estimated and the total transmission rate of the RTP session including retransmissions shall be adapted to the available link rate. The actual algorithms providing link-rate estimation and transmission-rate adaptation are implementation specific. Rules and information sources for the estimation of the available link rate are described in clause 10.2.1 of the present document. To adapt the total transmission rate including retransmissions, a PSS server can e.g. skip retransmissions, use the transmission rate adaptation described in clause 10.2.2 of the present document or use any other suitable method.

If the server uses multiple streams for rate adaptation, the server may receive retransmission requests for a stream that is different from the one it is currently using. The server should thus not flush its retransmission buffer after switching streams.

### 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];~~.~~
- 3GPP timed text format (see clause 7.9) RTP payload format according to [80];
- DRM encrypted RTP payload format according to clause K.1 in Annex K;
- RTP retransmission payload format according to [81].

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].

## 7.4 Video

If video is supported, ITU-T Recommendation H.263 ~~[22]~~ profile 0 level 45~~10~~ decoder [22][23] shall be supported. In addition, a PSS client should support:

- H.263 ~~[23]~~ Profile 3 Level 45~~10~~ decoder [22][23];
- MPEG-4 Visual Simple Profile Level 0b decoder~~,~~ [24] ~~and [25]~~.

The video buffer model 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.

## 7.7 Vector graphics

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

NOTE 1: The compression format for SVG content is GZIP [59], in accordance with the SVG specification [42].

NOTE 2: Adoption of SVG Tiny 1.2 to Release 6 is still being considered (as a working assumption) to PSS. Decision will be made as a late Release 6 item during TSG SA Meeting #26.

## 7.9 Timed text

If timed text is supported, PSS clients shall support [51]. ~~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.~~ Timed text may be transported over RTP or downloaded contained in 3GP files using Basic profile.

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;
- Basic profile and Progressive-download profile should be supported by PSS clients;
- Streaming server profile should be supported by PSS servers.

## 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 status feedback information free buffer space, next ADU to be decoded and playout delay. 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~~ NADU APP packets according to clause 5.3.3.5 and 6.2.3.2.

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 total buffer size, and the reported amount of free buffer space, the server can avoid overflowing the buffer. A server should assume that any sent RTP packet will consume receiver buffer space equal to the complete RTP packet size. For interleaved or aggregated media, the actual buffer space consumption may be slightly larger if buffering is done in the ADU domain. This is because each ADU may save metadata corresponding to the RTP header and payload fields, like timestamp and decoding sequence numbers individually. This should only be a problem if a server tries to fill exactly to the last free memory block.

The server can determine the time to underflow by calculating the amount of media time present in the buffer. This is done using the next ADU numbers and the highest received sequence number combined with the server's view of the sent ADUs and their decoding order and playout time. The information about the ADUs for 3GP files that are produced according to the streaming-server profile can be read from the "3gau" box [50]. It is also possible to derive some of the information about the ADUs from the media track, or hint-track, or the actual RTP packets.

~~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 value may improve~~s~~ 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. However care must be taken, to make correct use of the playout delay value as some of it is due to actual decoding delay, rather than post decoding buffering. Also the delay is only valid for the ADU actually reported on, and if that ADU has delayed playout, in regards to near-lying ADUs in the decoding order then an overestimation would occur.

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.

## 10.3 Issues with deriving adaptation information (informative)

This clause attempts to provide some insight into the functions and issues that exist in deriving client's buffer status in the server. The issues and the complexity of the functions depend on the media format, but can be characterised by media properties, in particular how much flexibility the media formats allows in transmission, decoding, and playout order. As there are three orderings of encoded media data that are possible, there are two re-orderings:

a) Data may be interleaved (i.e. the transmission order of data differs from the decoding order), and it must be de-interleaved before passing to the decoder.

b) There are forward references in the encoding, e.g. in a video stream, then those references are decoded 'early' (out of order) compared to playout order. Thus, the playout order in this case differs from the decode order. Thus having a playout order that may be different than decoding order.

In buffer management, we are trying to ensure

1. that the client's receiver buffer does not get over-filled;

2. that data does not arrive at an operation point after its need. Specifically, this means that ADUs should not be placed into the final playout queue with a timestamp that has already been passed in playout (this is under-run).

The parameters supplied enable a server to deduce at least this much. The server can always protect against buffer over-run by respecting the 'free space' that is periodically signalled by the client. This free-space is totalled over all data held before the decoder (decoder and de-interleave buffers). If the server desires more visibility, it can inspect the ADU that has been reported as 'next to decode'. If there has been no interleaving, the client holds all data between that ADU and the highest sequence number received, and will probably hold up to the last packet the server has sent. If interleave is used, then there may have been ADUs that were sent **after** the reported ADU, but which passed out of both the de-interleaving and decoder buffers before that ADU. The server would have to analyze the de-interleave process to work out which ADUs these are. The hint-track extension "3gau" to the 3GP file format [50] provides extended information about both the decoding and playout order in relation to transmission order of the ADUs. This extension does also provide the size of the ADUs to the server.

Protection against under-run is more subtle. It is in general not possible for the client to know which ADUs that are yet to be decoded (or yet to be received) that have earlier timestamps than ADUs already received and decoded. Therefore the client does not in fact know what is the 'latest playable timestamp', up to which it has received all the ADUs in the sequence to that time.

If the server does not adapt its transmission bit-rate and the transmission path has sufficient bit-rate, the parameters supplied at stream setup (such as the initial buffering delay) are sufficient to protect against under-run. The simple generalization of this is that if the server calculates its average bit-rate since starting the stream, and ensures that the average never falls below the bit-rate that would have been used without rate adaptation, it must be safe. Put in another way, the server may send a packet earlier than it would without rate-adaptation, but it might not be safe to send it later.

A more subtle analysis uses the reported information about the next-to-be-decoded ADU: the sequence number of the packet that contained it, the ADU number within that packet, and the offset (playout delay) of its timestamp (playback time) from the current playback time. Given the first pair of numbers, the server can find the ADU and therefore its timestamp. By subtracting the reported play-out delay from this timestamp, the server can now estimate the current playback time. It can find the earliest timestamp in the ADUs it has yet to transmit, and it can also examine the data that has been sent that will still be in the de-interleave buffer, for the earliest timestamp still held in the client's de-interleave buffer. If the earlier of these two timestamps is at, or close to, the current play time, the client has, or is about to, under-run.

Consider now the following cases, in order of complexity:

1. simple data that is neither interleaved nor re-ordered for display (e.g. AMR without interleave, AAC, H.263, MPEG-4 video).

2. data that is interleaved, but not re-ordered (e.g. AMR with interleave).

3. data that is re-ordered, but not interleaved (AVC without interleave).

4. data that is both interleaved and re-ordered (AVC with interleave).

Consider now over-run and under-run protection for these streams. In all cases, the free-space can be used to protect against over-run, and the maintenance of the average rate at or above the static rate protects against under-run.

1. By subtracting the reported free-space from the overall buffer size (reported in stream setup) the buffered data can be calculated. If this is nearly exhausted, the buffer is about to under-run. However for codecs with variable bit-rate encoding, the buffered space may represent different amount of playout time. In these cases the playout time present in the yet to be decoded part of the buffer can easily be calculated as the RTP timestamp difference between the latest ADU received by the client as reported implicitly by Highest Received Sequence number and the ADU reported by NADU.

2, The server can estimate the playback time as above. However to perform the calculation of the playout time of the buffer before the decoding, the server may need to maintain a list of the ADUs in the decoding order, rather than in transmission order. Also the data present in the de-interleaving buffer is not complete and would have holes in it and should not be considered to be playable. The server can determine, by looking at the decoding order of the different ADUs present in the transmitted packets, how far the client is expected to have a receiver buffer without holes, due to not yet transmitted packets.

3, In this case it is fairly complicated to estimate the actual playout time of the un-decoded media. The reason is that the present RTP timestamp associated with the ADUs may fluctuate widely in ADUs consecutive in both transmission and decoding order, due to the early decoding of referenced ADUs. Therefore to perform an accurate estimate the server needs to make special consideration of any ADU with early decoding so that it does not skew the measurement.

4, As 3 above, but with the further consideration of needing to perform any investigation in decoding order and consider the holes of the de-interleaving buffer.

## 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. Among the QoE metrics, corruption duration, successive loss of RTP packets, frame-rate deviation and jitter duration are of media level, whereas initial buffering duration and rebuffering duration are of session level.

### 11.2.5 Frame rate deviation

Frame rate deviation indicates the playback frame rate information. Frame rate deviation happens when the actual playback frame rate during a reporting period is deviated from a pre-defined value.

The actual playback frame rate is equal to the number of frames played during the reporting period divided by the time duration, in seconds, of the reporting period.

The parameter FR that denotes the pre-defined frame rate value is used with the "Framerate_Deviation" parameter in the "3GPP-QoE-Metrics" header. The value of FR shall be set by the server. The syntax for FR to be included in the "Measure-Spec" (clause 5.3.2.3.1) is as follows:

FR = "FR" "=" 1*DIGIT "." 1*DIGIT

The syntax for the Metrics-Name "Framerate_Deviation" 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. The absence of an event can be reported using the space (SP).

For the Metrics-Name "Framerate_Deviation", "Value" field indicates the frame rate deviation value that is equal to the pre-defined frame rate minus the actual playback frame rate. This metric is expressed in frames per second, and can be a fractional value, and can be negative. The metric value can occur only once for this metric.

### 11.2.6 Jitter duration

Jitter happens when the absolute difference between the actual playback time and the expected playback time is larger than a pre-defined value, which is 100 milliseconds. The expected time of a frame is equal to the actual playback time of the last played frame plus the difference between the NPT time of the frame and the NPT time of the last played frame.

The syntax for the Metrics-Name "Jitter_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 "Jitter_Duration", the "Value" field in 5.3.2.3.2 indicates the time duration of the playback jitter. The unit of this metrics is expressed in seconds, and can be a fractional value. There is the possibility that jitter occurs more than once during a reporting period. In that case the metric value can occur more than once indicating the number of jitter events.

The optional "Timestamp" field indicates the time when the jitter has occurred since the beginning of the reporting period. The value of the "Timestamp" is equal to the NPT time of the first played frame in the playback jitter, relative to the starting time of the reporting period.

## 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 |
| | | 3GPP-Integrity-Key | ND | O |
| | | 3GPP-SDP-Auth | 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 |
| | | 3GPP-SRTP-Config | ND | O |
| | | rtcp-fb | O | 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.

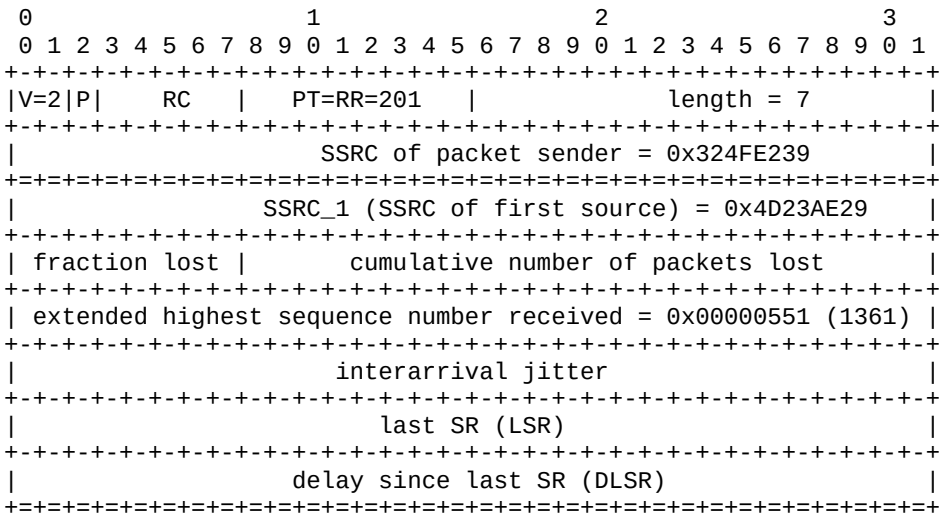
Note 5: The "framesize" attribute is only required for H.263 streams.

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.

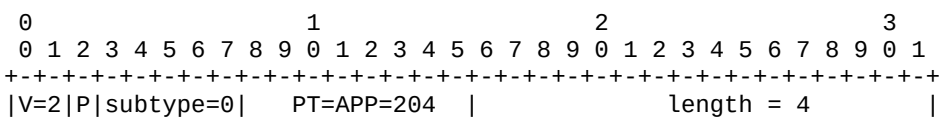
## A.3.3 Examples of RTCP APP packets for client buffer feedback

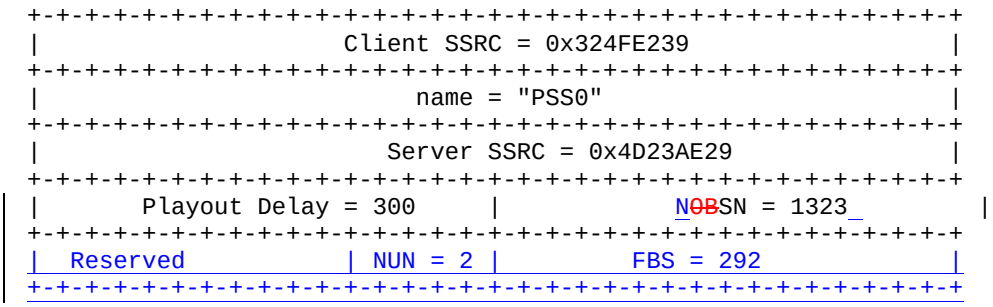
Example 1: The RTCP Receiver Report and ~~OBSN~~ NADU packet while having a number of packets for a single source in the receiver buffer and signalling the playout delay for the ~~oldest packet~~next unit to be decoded.

RTCP Receiver Report:




APP packet:






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~~ is able to derive all the ADUs that are in the receiver buffer by looking up all the ADUs it has sent which follow in decoding the second unit of packet SN 1323 and which were sent up to packet 1361. The total buffer size is~~of~~ 35000 bytes as indicated during the RTSP session setup (see rate-adaptation example in clause A.2.1). The available free space in the buffer is report as 292 64-byte blocks, which equals 18688 bytes of free buffer space.

~~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).~~The server is able to measure the time difference between the next ADU to be decoded and the next ADU it will send by comparing the decoding times of these units. Depending on this value, it is able to adapt using e.g. bitstream switching or bitstream thinning.

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.

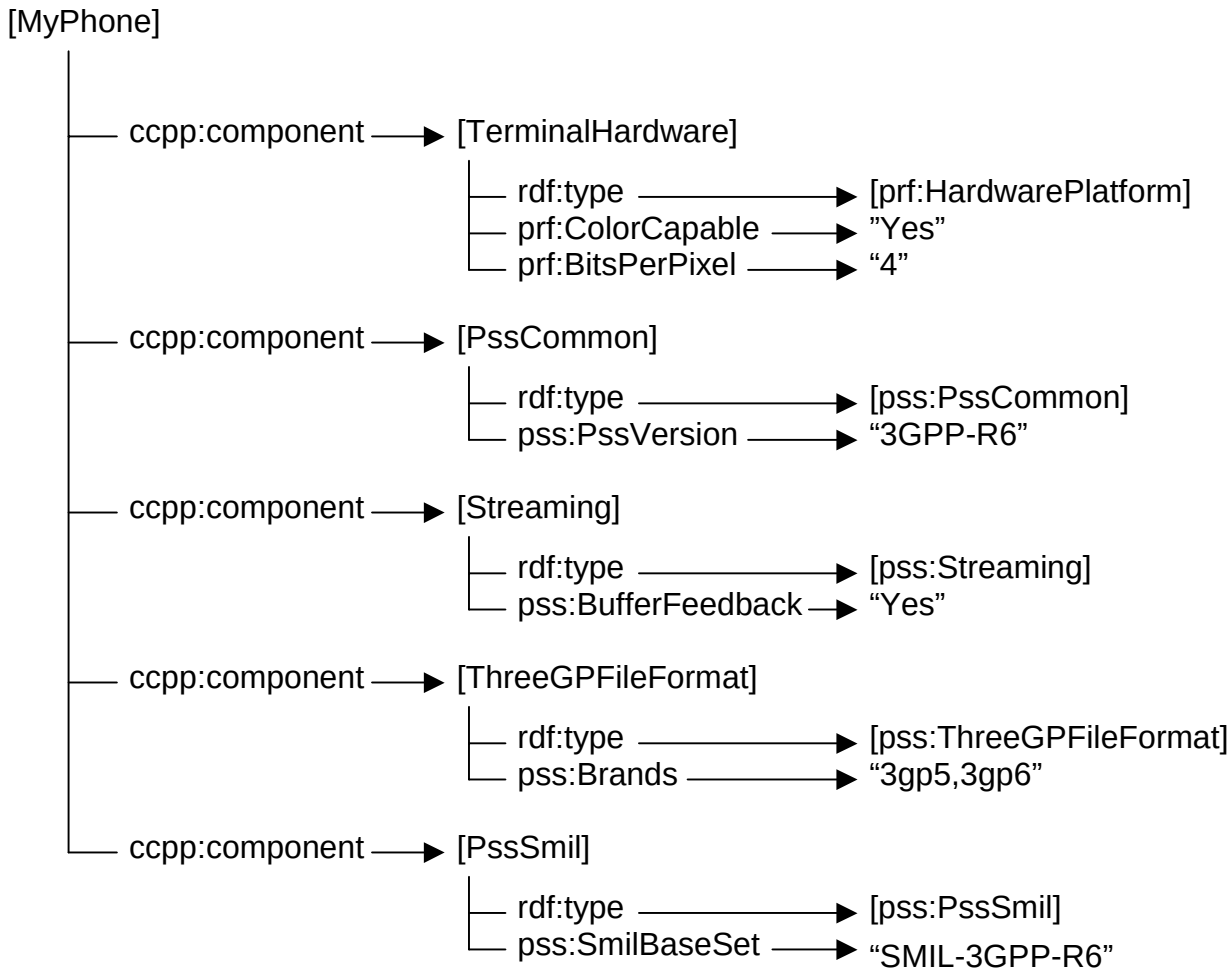
Example 2: Reporting an empty buffer.

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-~~NADU packet would carry an ~~OBSN-~~NSN 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.4.3 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.


[MyPhone]
ccpp:component
[TerminalHardware]
rdf:type
[prf:HardwarePlatform]
prf:ColorCapable
”Yes”
prf:BitsPerPixel
“4”
ccpp:component
[PssCommon]
rdf:type
[pss:PssCommon]
pss:PssVersion
“3GPP-R6”
ccpp:component
[Streaming]
rdf:type
[pss:Streaming]
pss:BufferFeedback
“Yes”
ccpp:component
[ThreeGPFileFormat]
rdf:type
[pss:ThreeGPFileFormat]
pss:Brands
“3gp5,3gp6”
ccpp:component
[PssSmil]
rdf:type
[pss:PssSmil]
pss:SmilBaseSet
“SMIL-3GPP-R6”

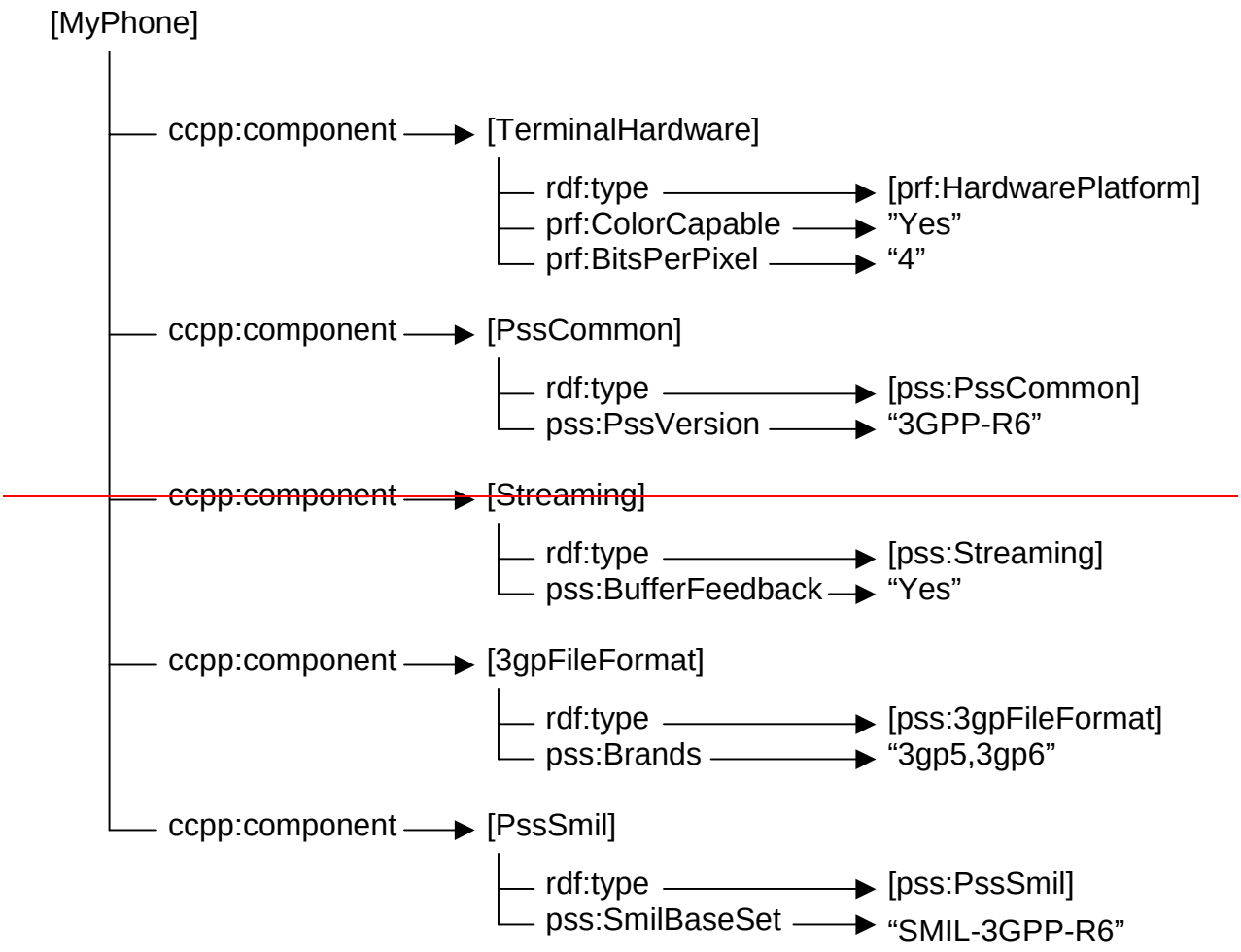



**Figure A.2: Illustration of the profile structure**

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), **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 namespace. (URI http://www.3gpp.org/profiles/PSS/ccppschema-PSS6).

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.4.7 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" />
        <pss6:3gppThreeGPPLinkChar>Yes</pss6:3gppThreeGPPLinkChar>
        <pss6:AdaptationSupportBufferFeedback>Yes</pss6:AdaptationSupportBufferFeedback>
        <pss6:ExtendedRtcpReports>Yes</pss6:ExtendedRtcpReports>
        <pss6:MediaAlternatives>Yes</pss6:MediaAlternatives3gppLinkChar>
        <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=4510</rdf:li>
            <rdf:li>video/H263-2000;profile=3;level=4510</rdf:li>
            <rdf:li>video/MP4V-ES</rdf:li>
          </rdf:Bag>
        </pss6:StreamingAccept>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID="3gpThreeGPFileFormat">
      <rdf:type rdf:resource=" http://www.3gpp.org/profiles/PSS/ccppschema-
PSS6#3gpThreeGPFileFormat" />
        <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>
        <pss6:3gpThreeGPAccept>
          <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=4510</rdf:li>
            <rdf:li>video/H263-2000;profile=3;level=4510</rdf:li>
            <rdf:li>video/Timed-Text</rdf:li>
          </rdf:Bag>
        </pss6:3gpThreeGPAccept>
      </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>
```

# 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="#3gpThreeGPFileFormat"/>
    <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, 3gpThreeGPFileFormat 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="3gpThreeGPFileFormat">
    <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: 3gpThreeGPFileFormat</rdfs:label>
    <rdfs:comment>
      The 3gpThreeGPFileFormat 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 capabibilites regarding streaming and 3GP files that are
      part of a SMIL presentation are expressed by the vocabularies specified by
      the Streaming and 3gpThreeGPFileFormat 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: LiteralNumber
      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: LiteralNumber
      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: LiteralNumber
      Resolution: Locked
      Examples: "Yes"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="RtpRetransmission">
    <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 RTP
      retransmission according to clause 6.2.3.3 of the specification.
      Legal values are "Yes" and "No".

      Type: Literal
      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: LiteralNumber
      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="StreamingOmaDrm">
    <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: Indicates whether the device supports streamed OMA DRM
      protected content, as defined by OMA and Annex K. Legal values are OMA
      Version numbers supported as a floating number. 0.0 indicates no support.

      Type: Literal (bag)
      Resolution: Locked
      Examples: "2.0"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="PSSIntegrity">
    <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: Indicates whether the device supports integrity protection
      for streamed content as defined by Annex K.2. Legal values are "Yes" and
```

```
      "No".

      Type: Literal
      Resolution: Locked
      Examples: "Yes"
    </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 168000. According to Annex G, 168000 is the default peak
      decoding byte rate for the mandatory video codec profile and level
      (H.263 Profile 0 Level 4510). Legal values are integer values greater than
      or equal to 168000.

      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: 3gpThreeGPFileFormat ***** -->

  <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="#3gpThreeGPFileFormat"/>
```

```
    <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="3gpThreeGPAccept">
    <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="#3gpThreeGPFileFormat"/>
    <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=4510,audio/AMR,Timed-text"
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="3gpThreeGPAccept-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="#3gpThreeGPFileFormat"/>
    <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 3gpThreeGPAccept-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 3gpThreeGPAccept-Subset, this means that
      3gpThreeGPAccept-Subset has precedence over 3gpThreeGPAccept. 3gpThreeGPAccept shall always
      include the corresponding content types for which 3gpThreeGPAccept-Subset
      specifies subsets of. No legal values are currently defined.

      Type: Literal (bag)
      Resolution: Locked
    </rdfs:comment>
  </rdf:Description>

  <rdf:Description ID="ThreeGPOmaDrm">
    <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="#ThreeGPFileFormat"/>
    <rdfs:comment>
      Description: List of the OMA DRM versions that is supported to be used
      for DRM protection of content present in the 3GP file format. Legal values
      are OMA DRM version numbers as floating values. 0.0 indicates no support.

      Type: Literal (bag)
      Resolution: Locked
      Examples: "2.0"
    </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 3gpThreeGPFileFormat
      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>
```

# 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 45~~10~~ requires support for bit-rates up to ~~64~~128000 bits per second. Thus, the peak decoding byte rate equals to ~~8~~16000 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 Profile 0 Level 45~~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 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
```

# Annex K (normative): Digital rights management extensions

This annex specifies extensions to support Open Mobile Alliance (OMA) digital rights management (DRM) version 2 [74]. The first extension is an RTP payload format that enables confidentiality protection of individual RTP payloads

used in a streaming session. The second extension defines the necessary key management and protocol support for the optional integrity protection of RTP payloads using SRTP [72] between streaming server and client.

# K.1 RTP payload format for encryption

This clause defines an RTP payload format for confidentiality protection for OMA DRM version 2 [74] for streamed media within PSS. The format specification addresses the following requirements:

- Support random seek capabilities in the encrypted media stream;
- Support pre-encryption of RTP payloads for usage in RTP hint-tracks as present in the 3GPP file format [50];
- Support selective encryption of individual payloads;
- Support usage of a strong encryption mechanism;
- Support arbitrary media payload formats.

To fulfil the above requirements a solution based on an RTP payload format that encapsulates an original RTP payload into a new RTP payload has been developed. The complete original payload is encrypted using a crypto transform. This specification defines one crypto transform using AES [77] in counter mode with a 128-bit key. To enable pre-encryption and random seek capabilities, an explicit Initialization Vector sequence number (IVSN) is used to derive the real initialization vector (IV). A minimalistic approach is taken in regards to overhead, and therefore the RTP payload type is used to support selective encryption, provide indication of the original RTP payload and determine any protection configuration. Thus there is need for a number of parameters to be signalled in relation to any defined payload type using this format.

To be able to use any other crypto transform one will need to identify if the IVSN field is needed, or some other field(s) are needed in addition to the encrypted body, and define these. To indicate this new transform, a new MIME subtype is defined that identifies the crypto transform used. Such a crypto transform could also define the presence of key indicator fields.

The description of the RTP payload format below uses the following definitions:

Content Encryption Key (CEK): The key used to encrypt the content, i.e. the original payloads.

Encrypted body: The encrypted bits of an original payload.

Encryption payload format: The RTP payload format defined in this chapter.

Encryption payload: The RTP payload that consists of an IV sequence number, key indicator field, and an encrypted body.

Initialization Vector (IV): The starting state of the cryptographic mode.

Original payload: A complete RTP payload in accordance with another RTP payload format specification.

Original RTP packet: A complete RTP packet that contains header values and payloads in accordance with the RTP specification and another RTP payload format specification.

Protected RTP packet: An RTP packet with the encryption payload format as payload, and its header values set according to RTP and the encryption payload format.

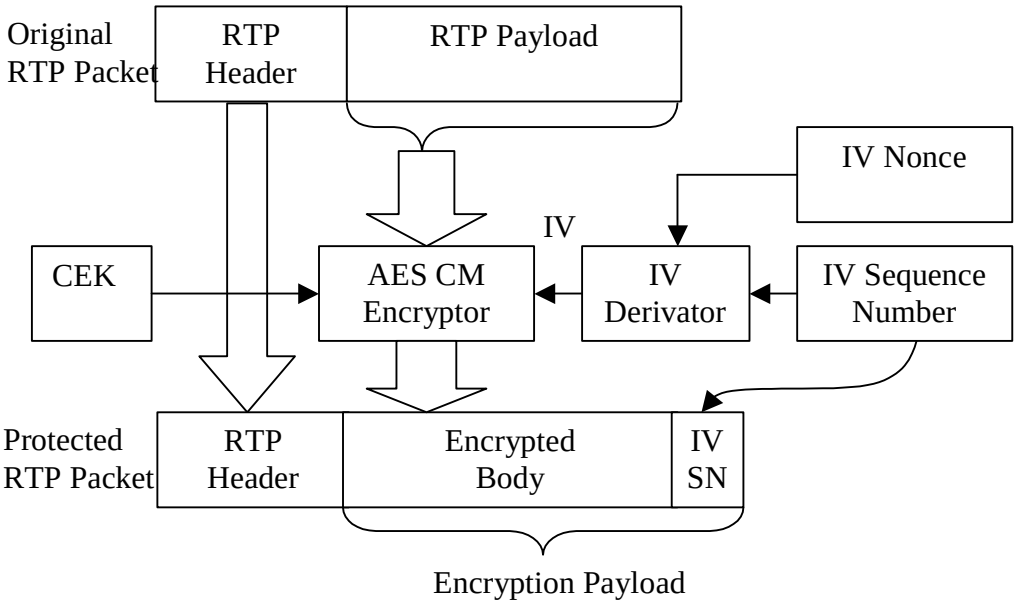



**Figure K.1: Schematic process - from an unprotected RTP packet to a protected one**

The confidentiality protection of the original RTP payload is accomplished through the encryption of the complete payload using a crypto transform, the defined format uses AES in counter mode (AES CM) with 12- bit keys, as shown in Figure K.1. The encryption of each individual payload is made independently from each other by assigning an Initialization Vector to each payload. In order to avoid sending the complete IV (128 bits for AES CM) in each RTP packet, a derivation process is used, to create the IV for each packet. As the IV derivation is a fully defined operation, the receiver can also perform it to determine the full IV. The IV Nonce is to protect against pre-computational attacks and is signalled out of band from the RTP stream. The IV sequence number used, as input to the IV derivation, is placed in the RTP payload of the protected packet together with the resulting ciphertext.

The header fields of a protected RTP packet are populated based on the RTP header fields of the original packet. The only field that is necessary to change is the RTP payload type, which is replaced with another type indicating that the RTP payload is using the encryption payload format. Further the payload type is also used to indicate which original payload type the packet contains. This usage of the payload type avoids using any bit in the RTP payload for the signalling.

No bits in the payload format need to be spent to enable the usage of selective encryption. This is also accomplished by using the payload type of the RTP header. A sender utilizing selective encryption, (on a packet-by-packet basis) signals for each packet if it wants to send the RTP payload protected or not, by using the corresponding payload type and format. A simple de-multiplexing as shown in Figure K.2 is all that is required on the receiver side to determine which payloads that needs decryption. A signalling attribute is defined to inform the receiver when selective encryption is used.

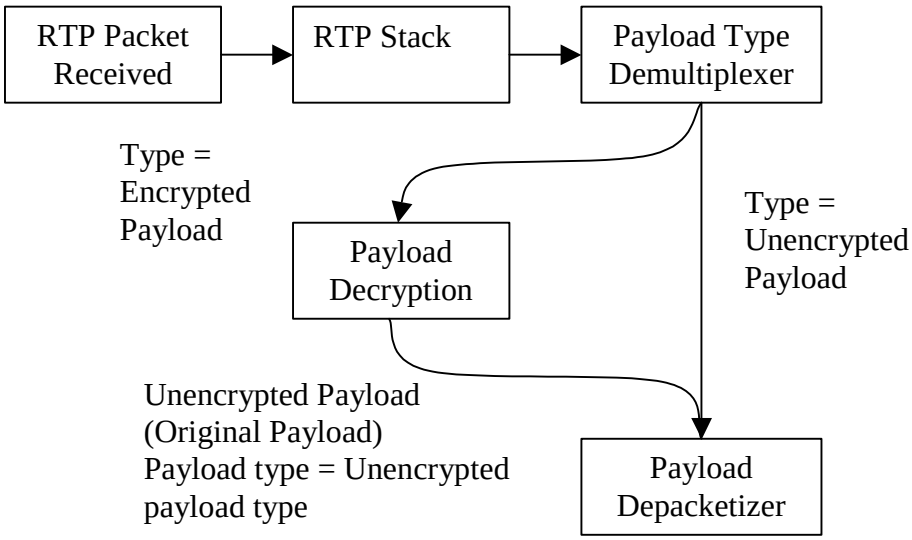



**Figure K.2: Flow for packet decryption including selective encryption.**

This payload format and its operations are based on OMA DRM version 2.0 [74], which includes specifications for DRM key management and how to declare the permissions and constraints governing the decoded media. The signalling provides DRM specific parameters, namely the DRM ContentID and the RightsIssuerURL, which points to the Rights Issuer from where a Rights Object corresponding to the content can be acquired. The security instantiation applies to complete RTP sessions and all media streams transmitted within it.

## K.1.1 Usage rules

One payload type shall be assigned for each original payload type that needs to be encrypted within each RTP session.

The same CEK shall be used for all RTP packets with the same payload type within a RTP session. The IV of each packet protected under a certain CEK must be unique, otherwise a two-time pad occurs (see below). This property must be ensured also if multiple sources are used across all the packets of the streams using the same CEK. Furthermore, if multiple encryption payload types are used they, may use the same CEK. In this case the uniqueness of the IV must hold over all the packets with different payload types using the same CEK. See also clause K.1.3.

The size of the added IV sequence number should be considered already in the creation of the original RTP payload. The added IVSN leads to a packet expansion of 4 bytes, which may result in a packet that is bigger than the MTU after the protection operation. This would lead to IP fragmentation, worse error robustness and increase the overhead. Thus the creator of the original payload should also take into consideration to make room for the extra bytes.

If authenticated signalling indicates that selective encryption shall not be used, then the receiver shall discard all RTP packets that contain payloads that are not encrypted.

## K.1.2 RTP payload format specification

This section specifies how to construct the binary format that is sent in the RTP payload and how the RTP header fields shall be assigned.

### K.1.2.1 RTP header usage

The RTP header usage depends on the original payload format, but is expected to be normal in accordance with RTP [9]. The value of any RTP header field shall be set in accordance with the definition for the original RTP payload format with the following exceptions or additions:

**Payload Type:** The RTP payload type for the encryption payload format shall be different from any payload type number assigned to an original payload type. The payload type number for an instance of the encryption payload format shall be bound to one and only one original payload format and its payload type number.

If the original payload uses non-standard definitions of the RTP header, the same considerations that apply to the processing of the RTP header of the original payload shall also apply to the encrypted payload format. If an original payload format does not define the usage of an RTP header field, then the RTP header field shall be used in accordance with RTP [9].

### K.1.2.2 RTP encryption payload

The RTP Encryption Payload shall consist of one encrypted body, followed by one Initialization Vector Sequence Number (IVSN). The two parts are defined as follows:

**Encrypted Body:** A variable length data block consisting of the encrypted original RTP payload. The encryption operation is performed as specified in clause K.1.3.

**IVSN:** A 4 bytes long field containing the initialization vector sequence number in network byte order.

## K.1.3 Encryption operations

Confidentiality of the encrypted RTP body is achieved by using an additive stream cipher, implemented by using the Advanced Encryption Standard (AES) cipher [77] run in counter mode to produce a keystream to encrypt/decrypt the original payload. Each original payload is encrypted with a distinct keystream segment, which is the concatenation of the 128-bit output blocks of the AES cipher in the encrypt direction, using the key CEK. The keystream is then bit-wise XORed with the original payload to create the encrypted body. Decryption is performed by the receiver in a similar way, XORing the encrypted body with the keystream to produce the original body.

The operation follows the definition and rules described in [72] for AES in counter mode, although the IV is defined as follows: IV = (nonce * $2^{16}$ ) XOR (IVSN * $2^{16}$)

(the above reconstruction of the IV from the IVSN is denoted as the IV Derivator in Figure K.3).

The 16 zeros in the least significant (right-most) bits of the IV are used as the counter, for generating the keystream needed to encrypt the payload.

IVSN is the 32-bit IV sequence number and is the only part of the IV to be explicitly carried in each packet.

The nonce is used against pre-computational attacks that are possible against stream ciphers. The nonce must be chosen randomly and independently and is sent to the client out-of-band (see section K.1.4). The length of the nonce shall be 112 bits, i.e. the IV nonce parameter shall be present and have a length of 112 bits prior to base 64 encoding. Before XOR:ing and "shifting" IVSN to form the above IV, an alignment with the nonce shall be made, considering also IVSN as a 112-bit value, by padding IVSN by 80 leading zeros.

The use of the IVSN and the nonce must be so that the IV of each packet protected under a certain CEK is unique, otherwise a two-time pad occurs causing the plaintext to leak (see [72]).

The use of the 16-bit inner counter fixes the maximum number of keystream blocks that can be generated for any fixed value of the IV to $2^{16}$, otherwise keystream re-use occurs compromising the security. Since AES has a block size of 128 bits, $2^{16}$ output blocks can generate $2^{23}$ bits of keystream (1048576 bytes), which are enough to encrypt the largest RTP packet (except if IPv6 jumbograms are used [76]).

The maximum number of packets that can be encrypted under the same CEK and for a given nonce is $2^{32}$ (due to the 32 bit IVSN).

This payload specifies security functionality for achieving confidentiality protection of RTP payloads. Because the RTP header is not protected, the inter-packet synchronization, payload types, and sequence ordering of the RTP packets are all examples of information that is not protected. The confidentiality of the encrypted original payload is depending on the strength of AES in counter mode with a 128-bit key and the utilized key management.

Not using integrity protection combined with an additive stream cipher like AES CM, may allow an attacker to purposefully and in a controlled fashion invert individual bits' values. If, in addition, an attacker knows the value of a certain bit in the RTP payload, it can change this bits value although it is encrypted, by a simple XOR of the encrypted bit with 1. Using integrity protection in conjunction with AES counter mode enables the client to detect such attacks on the cipher.

When using selective encryption, unencrypted packets disclose their content to anybody. Further, in case of lack of integrity and replay protection, it makes attacks that replay and modify the content extremely simple to perform [73]. Thus, integrity protection is strongly recommended if selective encryption is used. It is also recommended to integrity protect the flag indicating the presence of selective encryption (e.g. as described in section K.2), otherwise an attacker can tamper with it and turn the function on, allowing for the risks described above.

## K.1.4 Signalling

This clause specifies the RTP payload format MIME type, and how it is utilized in SDP. An example is included as well.

Any unknown MIME parameter shall be ignored.

### K.1.4.1 MIME type definition

MIME media type name: audio, video, text, application, image

MIME subtype name: rtp.enc.aescm128

Required parameters:

**opt:** The payload type number of the payload type contained in the encrypted payload. An integer value between 0-127.

**rate:** The timestamp rate of this payload type, which shall be the same as that of the original payload type. This is an integer value between 1 and 2^32.

**ContentID:** The OMA DRM content ID [75] used to identify the content when establishing a crypto context. The value is an RFC 2396 [60] URI, which shall be quoted using <">.

**RightsIssuerURL:** The right issuer URL as defined by OMA DRM [75]. The value is an URI in accordance with RFC 2396 [60], which shall be quoted using <">.

**IVnonce:** The value of this parameter is the nonce that forms the IV as specified by the crypto transform, encoded using Base 64 [69].

Optional parameters:

**SelectiveEncryption:** Indicates if this stream is selectively encrypted. Allowed values are 0 (false) and 1 (true). If not present, selective encryption shall not be used. Please note that unless this indicator is integrity protected, it fulfils no purpose. Encoding considerations:

This type is only defined for transfer via RTP (RFC 3550).

Security considerations:

See considerations raised in RTP RFC 3550 [9] and any applicable profile like RFC 3551 [10] or RFC 3711 [72]. Further see 3GPP TS 26.234, Release 6, Annex K for comments on security issues. The main issues that exists are:

- This RTP payload format only confidentiality protects the RTP payload, thus header information is leaked, similarly to SRTP.

- The use of stream ciphers as AES CM and no integrity protection allows an attacker to purposefully attack the content of the encrypted RTP payload by switching individual bits.

- The usage of selective encryption without integrity protection allows for an attacker to perform any replacements of complete RTP payloads and packets it desires.

- The payload format makes the receiver vulnerable to denial of service attacks that inserts RTP packets into the stream, that the receiver then interprets as being encrypted thus wasting computational resources. To prevent this attack, authentication needs to be used.

Interoperability considerations:

Published specification:

3GPP TS 26.234, Release 6.
Open Mobile Alliance DRM Content Format V2.0

Applications which use this media type:

Third Generation Partnership Project (3GPP) Packet-switched Streaming Service (PSS) clients and servers, which supports the Open Mobile Alliance's specification of Digital Rights Management version 2.0.

Additional information:

Magic number(s): N/A

File extension(s): N/A

Macintosh File Type Code(s): N/A

Person & email address to contact for further information:

magnus.westerlund@ericsson.com

Intended usage:

Common

Author/Change controller:

3GPP TSG SA

## K.1.4.2 Mapping of MIME to SDP

The MIME media types for the encrypted RTP payload format and its parameter strings are mapped to fields in the Session Description Protocol (SDP) [6] as follows:

- The media name in the "m=" line of SDP shall be set to the used media type, i.e. audio, video, text, application, or image.
- The encoding name in the "a=rtpmap" line of SDP shall be rtp.enc.aescm128 (the MIME subtype).
- The clock rate in the "a=rtpmap" line shall be equal to the rate parameter.
- The remaining parameters when present, shall be included in the "a=fmtp" line of SDP. These parameters are expressed as a MIME media type string, in the form of a semicolon separated list of parameter=value pairs.

Note that the payload format (encoding) names are commonly shown in upper case. MIME subtypes are commonly shown in lower case. These names are case-insensitive in both places. Similarly, parameter names are case-insensitive both in MIME types and in the default mapping to the SDP a=fmtp attribute.

This MIME type is only intended for declarative usage, like in RTSP. The usage and behaviour in the SDP Offer/Answer model is undefined.

## K.1.4.3 SDP example

```
v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video including DRM
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 98
b=AS:13
b=RR:350
b=RS:300
a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align=1
a=rtpmap:98 RTP.ENC.AESCM128/8000
a=fmtp:98 opt=97; ContentID="content1000221@ContentIssuer.com";
RightsIssuerURL="http://drm.rightsserver.org/1000221";
IVnonce=JDE0SYJCAAqWUwWJiBM=; SelectiveEncryption=1
a=control: streamID=0
a=3GPP-Adaptation-Support:2
m=video 0 RTP/AVP 99 100
b=AS:64
b=RR:2000
b=RS:1200
a=rtpmap:99 H263-2000/90000
a=fmtp:99 profile=3;level=10
a=rtpmap:100 RTP.ENC.AESCM128/90000
a=fmtp:100 opt=99; ContentID="content6188164@ContentIssuer.com";
RightsIssuerURL=" http://drm.rightsserver.org/6188164"; IVnonce=
IwOSRWeSAUiVEiN5gVA=
a=control: streamID=1
a=3GPP-Adaptation-Support:1
```

# K.2 Integrity protection of RTP

An integrity protection mechanism is defined to optionally protect the communication between the streaming server and the client. The mechanism uses the Secure Real time Transport Protocol [72]. SRTP can provide confidentiality of the RTP payload, and integrity protection (with replay protection) of the RTP packet. The confidentiality protection of the RTP payload may be done using the OMA DRM with the above specified payload format, hence the use of SRTP defined in this specification is only for integrity protection.

The assumed trust model for the integrity protection mechanism is that the streaming server is trusted (except for the possibility of accessing the content, if it is pre-encrypted). It is further assumed that the content distribution network, delivering content (and keys) from the content provider to the streaming server is secure.

## K.2.1 Integrity key exchange

The SRTP master key is generated by the streaming server based on an integrity key provided by the Content Provider. This assures the client that the streaming server has indeed a trusted relation with the Content Provider and is an "authorized" server. The server selects randomly nonce values per-session, so that the resulting SRTP master key(s) (derived from the Content Provider's integrity key and the server's nonces) have a per-session/per-client validity. An integrity key, similarly derived, is used to protect the SDP attributes as well. This is detailed in the following and illustrated in Figure K.3. One assumption is that the Content Provider and the client have exchanged a pre-shared key (denoted CEK hereby) in advance. This specification uses the OMA DRM version 2 specified content encryption key [74] as the shared key, and relies on the OMA DRM key management to deliver the CEK key to the receiver. Please note that an OMA content protection key may be produced for only the purpose of protecting the integrity key, and not be used for confidentiality protection of the content when streaming.

If integrity protection of the content object is required between the streaming server and the client, the Content Provider generates a 160-bit integrity key k for the content object. The content object (possibly pre-encrypted under the correspondent CEK, see section K.1) is then sent to the streaming server. The Content Provider also sends the key k and a copy of it encrypted under a content object's CEK. The encrypted copy of k can be decrypted only by the clients who possess the right CEK (signalled by the content identifier that accompanies the encrypted k in the SDP attribute). To encrypt the key k under the CEK, the AES key wrap method is used, as specified in [78]. The default IV shall be used, as specified in [78]. (Note: key wrap wants the protected k to be multiple of 64 bits. The key k is here requested to be 160 bits, so that length is defined. Pad the key to be multiple of 64, e.g., with zeros to a length of 192 bits; the padding will be discarded at the receiver anyway).

To avoid that multiple clients share the same session key material, the streaming server randomly and independently generates a 128-bit i_nonce value per RTP session. The streaming server derives two keys from k and i_nonce values:

1) a key Ks, to integrity protect the SDP description (see section K.2.2) including the security parameters needed to setup SRTP for the media protection. This includes protection of the flag indicating if selective (pre-)encryption (section K.1) is used, which (in absence of integrity protection) could otherwise be tampered (i.e. by modifying it, an attacker can turn on selective encryption, opening to the risks described in section K.1.3).

2) an SRTP master key Km for each RTP session, for integrity protecting it (by applying SRTP, see section K.2.3).

The server then sends to the client the i_nonce values and the encrypted copy of k (together with a freshness token, whose usage is explained later), within the integrity protected SDP description.

Since the client knows the CEK, he can decrypt k. The client performs the key derivation, so that at this point he and the streaming server share the derived keys Ks and Km(s). The client further verifies the authenticity of the SDP part (section K.2.2.).

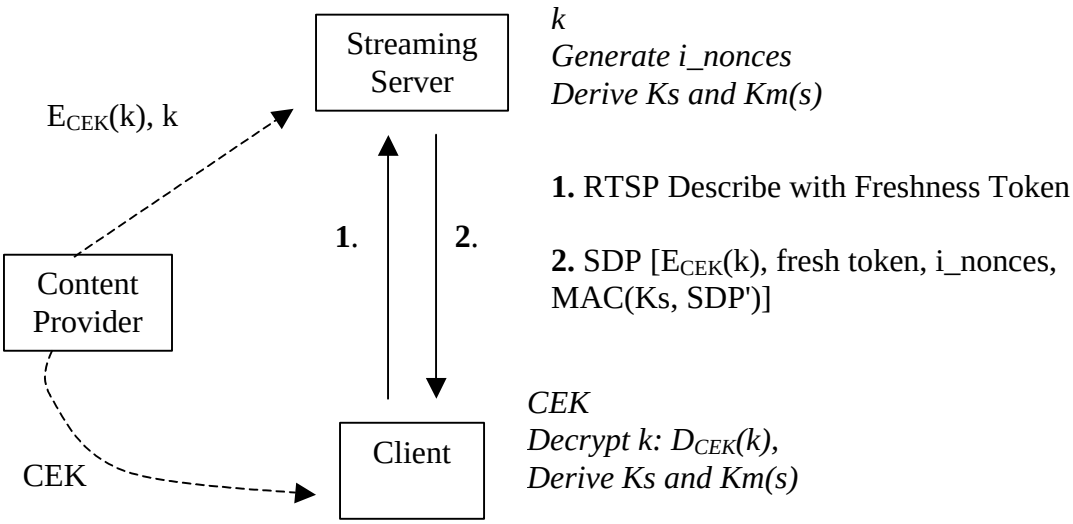



**Figure K.3: Key management scheme for SRTP to protect the streaming session. The RTP session is not shown, only the signalling (SDP). Terms in *italics* are present or calculated at the peer, terms in normal font are transmitted. Dotted line assumes trusted channels. The fresh(ness) token comes from the Describe Request message. SDP' is a subset of SDP.**

It is assumed that the key derivation of Ks and Km can be securely performed within the trusted area of the device (in this way, the key k is not leaked out of the trusted area of the device. Note that k is distributed to many clients, but the use of the i_nonce values (together with the freshness token, see below) is such that the derived keys for SDP and SRTP are tied to the particular session/client, hence it prevents possible manipulation/replay attacks from the other clients). Furthermore, the key Ks should not leave the device.

As it is not guaranteed that the keys Km remain within the trusted area of the device (while k and Ks are, by assumption), to avoid bad clients to misbehave by e.g. manipulating/replaying other clients' messages, a fresh token is generated within the trusted area of the device. When the client contacts the streaming server, he sends the fresh token within the RTSP Describe request. The streaming server shall place the received fresh token within the authenticated SDP that is sent from the server to the client in the Describe message. A trusted device that receives an authenticated SDP without a proper, previously generated fresh token, shall abort the connection setup. The fresh token is a randomly and independently generated 128-bit token. A produced fresh token shall be consumed once by the trusted device, and then erased.

Note that the Content Provider should distribute a different key k per server, unless the servers are trusted to act fairly to each other and the streaming clients (having the same k, and by observing an i_nonce sent by a server to a client, they can derive the related keys and perform any attack).

Each distributed i_nonce needs to be bound to the actual RTSP session and delivery where it is used. This is easiest accomplished using session specific RTSP control URIs. It will be the servers responsibility to handle this dynamic and temporarily created i_nonce and its corresponding control URI(s). The server will also need to prevent undesired reuse of any i_nonce, see K.2.4.2.

## K.2.2 Security parameters exchange

This clause defines three SDP attributes, one to transport the freshness token, the encrypted key, and the related information, one to carry the SRTP configuration and the integrity nonce, and a third to integrity protect some important fields in the SDP. An RTSP header to carry the Freshness Token in Describe requests is also defined.

Common ABNF [53] definitions are:

Token = 1*( %x21 / %x23-27 / %x2A-2B / %x2D-2E / %x30-39 / %x41-5A / %x5E-7A / %x7C / %x7E )

base64 = *base64-unit [base64-pad]

base64-unit = 4base64-char

base64-pad = 2base64-char "==" / 3base64-char "="

base64-char = ALPHA / DIGIT / "+" / "/"

### K.2.2.1 SDP integrity key information attribute

The protected key k (together with any information necessary to identify the CEK key used to protect k) and the freshness token are carried in the session level SDP attribute "a=3GPP-Integrity-Key".

The ABNF [53] for the session level attribute is defined as:

3gpp-integrity-attribute = "a=" "3GPP-Integrity-Key" ":" enc-intg-key [SP fresh-token]

fresh-token = base64

enc-intg-key = key-method ":" [keydata]

key-method = "OMADRMv2" / key-method-ext

key-method-ext = token

keydata = 1*VCHAR / OMADRMv2-keydata

OMADRMv2-keydata = omadrm-enc-key-data "," content-id-uri "," right-issuer-url

omadrm-enc-key-data = base64

content-id-uri = DQUOTE absoluteURI DQUOTE

right-issuer-url = DQUOTE absoluteURI DQUOTE

absoluteURI = as defined by RFC 2396 [60]

enc-intg-key is the encrypted key k, carried as defined by the method identifier. For the key distribution method "OMADRMv2" the base64 encoded encrypted key data is carried together with the corresponding content ID URI and rights issuer URL used to identify the CEK. When using the "OMADRMv2" keying method the following definition of the keydata applies:

omadrm-enc-key-data = BASE64(AES(CEK,k))

To encrypt the key k under the CEK, the AES key wrap method shall be used, as specified in [78]. The default IV shall be used, as specified in [78]. The key k is 160 bits and shall be padded to 192 bits, prior to wrapping. The output will be a 256 binary value, which shall be base64 encoded. The CEK is identified through the ContentID URI present. The rights object can be acquired from the location indicated through the rights issuer URL.

The freshness token (fresh-token) shall be a base64 encoded 128-bit binary value.

The attribute may also be used without any key data and freshness token, to indicate that this specification and its key method shall be used for key management. A client receiving a SDP without a freshness token shall when desiring to set up a session include a freshness token in a RTSP DESCRIBE and request a new SDP using the session level RTSP control URI present in the received SDP.

### K.2.2.2 SDP SRTP configuration attribute

The SRTP specific nonce, SRTP salt key, and any SRTP configuration information are carried in a media level SDP attribute "a=3GPP-SRTP-Config".

3gpp-integrity-attribute = "a=" "3GPP-SRTP-Config" ":" intg-nonce SP srtp-key-salt *SRTP-session-param

intg-nonce = base64

srtp-key-salt = base64

srtp-session-param = SP srtp-param "=" 1*VCHAR

srtp-param = "auth-tag-len" / srtp-param-extension

srtp-param-extension = token

The "srtp-key-salt" shall be the base64 encoding of the 112 bits of SRTP salt key. The "intg-nonce" shall be the base64 encoding of the 128 bits of "i_nonce".

The SRTP session parameter "auth-tag-len" shall be present to indicate the used SRTP authentication tags length. Valid values are 32 or 80.

### K.2.2.3 SDP authentication attribute

Parts of the SDP description are integrity protected using a message authentication code (MAC). A new session level SDP attribute "a=3GPP-SDP-Auth" carries the 160-bit MAC that is calculated as:

auth-tag = HMAC-SHA1 (Ks, m)

Ks is a 160-bit key taken from the output of HMAC-SHA1, calculated over k, and i_nonce concatenated to the label "SDP_integrity_key":

Ks = HMAC-SHA1 (k, i_nonce || "SDP_integrity_key")

The coverage of the MAC (m) is defined below. The i_nonce value fed into the above HMAC is the i_nonce value carried in the first media description (from a m= line until next) of the correspondent SDP description.

Both the server and the client can calculate Ks because they possess k (and the client receives i_nonce from the server). The k is available through the session SDP attribute "a=3GPP-Integrity-Key". Hence the client needs first to extract the fields from the SDP, decrypt k, derive all the keys, and only after can verify the validity of the SDP MAC. If the verification is unsuccessful, the complete session setup operation shall be aborted.

The message to perform the authentication over (m) is created in the following way from the SDP:

1. Create the SDP (S) without the "a=3GPP-SDP-Auth" attribute.
2. m is any empty string.
3. Start at the first line of S.
4. Check if the line contains any of the following SDP fields or attributes:
   - m=
   - a=control
   - a=fmtp
   - a=rtpmap
   - a=3GPP-Integrity-Key
   - a=3GPP-SRTP-Config

   If that is true, then add the complete line including the CRLF to the end of m.
5. Go to the next line in the SDP, and go to bullet 4, until end of S.

Thus forming m as an excerpt of the original SDP maintaining order of the selected fields. Which is then used to calculate the 160-bit integrity tag as specified above.

The ABNF [53] for the authentication attribute is:

3gpp-authentication-attribute = "a=" "3GPP-SDP-Auth" ":" 3gpp-auth-tag

3gpp-auth-tag = base64

The 3gpp-auth-tag shall consist of the base64 [69] encoding of the 160 bits of binary "auth-tag" defined above.

When calculated the attribute is added to the SDP at the session level.

### K.2.2.4 Freshness token RTSP header

To enable the client to supply the server with a freshness token, a new RTSP header is defined.

The ABNF for this header is:

Freshness-Token-Hdr = "3GPP-Freshness-Token" ":" LWS fresh-token

fresh-token = As defined in clause K.2.2.1

LWS = As defined in RFC 2326 [5].

The header may be included in RTSP DESCRIBE requests. A proxy shall not modify, or add this header. The header shall be included if the client has received indication that the integrity protection and the here specified key management are used. To potentially save a round trip a client may include the header and freshness token in any RTSP Describe request, although no indication that integrity protection has been given. This avoids having the server to send SDP without keying material to indicate the necessity of including a freshness token.

## K.2.3 Media security protocol

The security parameters exchanged within the SDP are used to secure the RTP streaming session between the streaming server and the client.

For each RTP session (i.e. each media description), the SRTP master key Km is taken from the 128 left-most bits of the output of HMAC-SHA1, calculated over k, and i_nonce concatenated to the label "SRTP_master_key":

Km = HMAC-SHA1 (k, i_nonce || "SRTP_master_key")

where i_nonce is the i_nonce value carried in the 3gpp-srtp-config attribute of the correspondent media description.

Both the RTP stream and the corresponding RTCP stream are integrity protected. Replay protection shall be turned on.

The additional security parameters exchanged within the SDP (salt key, authentication tag length) are used to populate the corresponding parameters in the SRTP cryptographic context. The remaining parameters are chosen according to normal procedure in [72], and default values are used. With the exception of the following:

- SRTP encryption transform shall be NULL.
- SRTCP encryption transform shall be NULL.

The session authentication key for the integrity protection of the RTP/RTCP session is securely derived from the SRTP master key Km by applying the SRTP key derivation function, as defined in [72]. The Message Authentication Code tag that is appended per packet is based on HMAC-SHA1 and has a truncated length of 80 or 32 bits for RTP (always 80 bits for RTCP).

## K.2.4 Servers and content

This clause defines how to indicate at the above defined key-management shall be used in 3GPP file files [50], and gives further rules regarding handling of content and media announcements.

### K.2.4.1 3GP file format extensions

A server may use the streaming-server profile of the 3GPP file format [50] to indicate that integrity protection shall be applied. If hinted content is intended to be integrity protected it shall be hinted using the SRTP hint track, as specified by clause 7.6 in [50]. To identify the above specified key management mechanism, the following definitions shall be used:

- The **SRTPProcessBox** identifies the algorithms applied: EncryptionAlgorithmRTP and EncryptionAlgorithmRTCP shall be equal to ENUL, IntegrityAlgorithmRTP and IntegrityAlgorithmRTCP shall be equal to SHM2.

- The **SchemeTypeBox** field "SchemeType" shall be set to "pssi" and the field "SchemeVersion" shall be set to 0x01. The field "SchemeURI" shall be null.

- When OMA DRM v2 is used to establish the shared key the SchemeInformationBox shall contain a OMADRMPSSIntegrityKeyMgmtBox.

The key management and protection operation needs to be configured with the information present in the OMADRMPSSIntegrityKeyMgmtBox, defined in Table K.1. The SRTP tag lengths to use for this media is indicated with the RTPIntegrityTagLen field. Further the integrity key k and its encrypted version is also provided. The information necessary to identify which CEK that has been used to protected k in the server to client transport is also included.

**Table K.1: OMADRMPSSIntegrityKeyMgmtBox**

| Field | Type | Details | Value |
|---|---|---|---|
| **BoxHeader**.Size | Unsigned int(32) | | |
| **BoxHeader**.Type | Unsigned int(32) | | 'odik' |
| **BoxHeader**.Version | Unsigned int(8) | | 0 |
| **BoxHeader**.Flags | Bit(24) | | 0 |
| RTPIntegrityTagLen | Unsigned int(32) | The length of the Integrity tag to be used to apply for each RTP packet specified by the SRTP hint track. | 32 or 80 |
| IntegrityKey | Unsigned int(8)[20] | The 128 bit Integrity key (k) in the clear. | |
| ProtectedIntegrityKey | Unsigned int(8)[32] | The confidentiality protected key k. | |
| OMADRMContentIDURI | Unsigned int(8)[] | The ContentID URI that identifies the CEK that has been used to protect "ProtectedIntegrityKey". The field contains a null terminated UTF-8 string. | |
| OMADRMRightsIssueURL | Unsigned int(8)[] | The rights issuer URL where rights for the CEK can be obtained. The field contains a null terminated UTF-8 string. | |

## K.2.4.2 Server handling

A PSS server implementing this integrity protection will need to bind a generated set of integrity nonce and SRTP key salts, to a client's request to setup the session. This binding shall be accomplished using per session specific URIs. By encoding an index in the control URIs at both media and session level, the server can bind a generated set of security parameters. When the client has received a particular SDP with its control URIs and security parameters, it will perform a RTSP SETUP using the attached control URIs, thus indicating for the server which security parameters should be used in the session. As the server will generate a new SDP with session individual parameters that require state at the server, there exist some risk for denial of service in this usage. Therefore a streaming server is only required to keep a created state for 3 minutes. To further mitigate the risk of denial of service attacks, the server may restrict the number of states being allowed to create in a given time interval, thus bounding the amount of resources required for this procedure. If a client requests to perform a RTSP SETUP using a state that has expired, the server is recommended to perform a 302 RTSP redirect response to another URI to indicate that the client shall retrieve a new SDP with a valid state.

As specified by PSS the client can acquire a SDP for a session in multiple ways, RTSP DESCRIBE, HTTP GET, WAP, or messaging. As the integrity protection requires per session specific parameters the usage of RTSP DESCRIBE becomes a requirement to ensure that unique parameters are provided to different clients. However this does not rule out that a SDP is distributed through other means. A server shall support redirecting clients requesting to SETUP a session using a URI pointing to a generic, already in use, or expired parameter state. With generic parameter state, is such a state that is generated, only with the purpose of redirecting clients to retrieve a unique session state.

To avoid that any set of session specific parameters are reused, more often than what will happen when the parameters are randomly selected, the following methods should be employed:

- Ensure usage of good pseudo random functions.

- Any state being or having been used shall not be allowed to be used by another client until randomly selected again.

## K.2.5 Example

This clause shows an example including the key management protocol for the content integrity protection between the streaming server and the client. First is an overview in the form of a flow diagram (see Figure K.4).

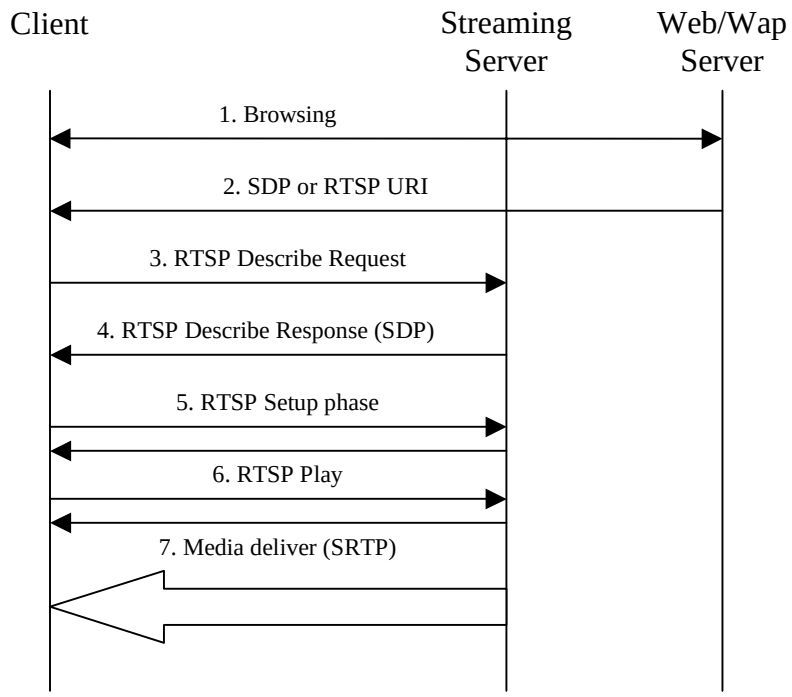



**Figure K.4: Flow diagram for Session Establishment with Integrity Protection**

1. (Optional) A user is browsing for streaming content.

2. (Optional) Upon finding interesting content the client retrieves either an RTSP URI or an SDP. If the client retrieves a SDP file, then that SDP will contain m= lines with RTP/SAVP and the integrity key management attributes. However the actual key related values will most probably not be used. See the following example:

```
v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video with DRM with confidentiality and Integrity protection.
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:rtsp://example.com/SecuredMedia/hobbs.3gp
a=3GPP-Integrity-Key: OMADRMv2:
m=audio 0 RTP/SAVP 97 98
b=AS:13
b=RR:350
b=RS:300
```

a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align=1
a=rtpmap:98 RTP.ENC.AESCM128/8000
a=fmtp:98 opt=97; ContentID=" content1000221@ContentIssuer.com";
RightsIssuerURL="http://drm.rightsserver.org/1000221"; IVnonce=JDE0SYJCAAqWUwWJiBM=;
SelectiveEncryption=1
**a=control:rtsp://example.com/SecuredMedia/hobbs.3gp/streamID=0**
a=3GPP-Adaptation-Support:2
**m=video 0 RTP/SAVP 99 100**
b=AS:64
b=RR:2000
b=RS:1200
a=rtpmap:99 H263-2000/90000
a=fmtp:99 profile=3;level=10
a=rtpmap:100 RTP.ENC.AESCM128/90000
a=fmtp:100 opt=99; ContentID="content6188164@ContentIssuer.com"; RightsIssuerURL="
http://drm.rightsserver.org/6188164"; IVnonce= IwOSRWeSAUiVEiN5gVA=
**a=control:rtsp://example.com/SecuredMedia/hobbs.3gp/streamID=1**
a=3GPP-Adaptation-Support:1

The client upon receiving this SDP can determine the need to support SRTP for this media (signalled by the SAVP profile). Also the key management scheme is evident, through the SDP attribute a=3GPP-Integrity-Key and its method identifier. The a=3GPP-Integrity-Key not containing key and freshness token also tells the client that it needs to request a new SDP containing session specific values.

3. The client may now know (due to the SDP) that it needs to retrieve a SDP from the streaming server. Therefore it sends an RTSP DESCRIBE request to the server including a freshness token.

**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**
**3GPP-Freshness-Token: zSARrvlkL94OcWB/yqDszw==**

4. The server has received a DESCRIBE request for content that shall be integrity protected. If the server is delivering content from a 3GP file, the server determines this based on the SRTP hint-tracks present in the file, and its schemeTypeBox. If this indicates that the key management to be used is the one specified above. The server generates the i nonce values, and derives the keys Ks and Km. The server specifies the SRTP security parameters within the SDP, adding the i nonce values, the encrypted copy of k, and the freshness token, and integrity protects such SDP part with the derived key Ks. This results in a new SDP looking like this:

v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video with DRM with confidentiality and Integrity protection.
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:rtsp://example.com/session0000012838984**
**a=3GPP-Integrity-Key: OMADRMv2: 1SCxWEMNe397m24SwgyRhg==,"**
**content1000221@ContentIssuer.com","http://drm.rightsserver.org/1000221"**
**zSARrvlkL94OcWB/yqDszw==**
**a=3GPP-SDP-Auth:1SCxWEMNe397m24SwgyRhg== fmVZNGmrsuVmyGIEtwVaU2xFwOw=**
**m=audio 0 RTP/SAVP 97 98**
b=AS:13
b=RR:350
b=RS:300
a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align=1
a=rtpmap:98 RTP.ENC.AESCM128/8000

a=fmtp:98 opt=97; ContentID=" content1000221@ContentIssuer.com"; RightsIssuerURL="
http://drm.rightsserver.org/1000221"; IVnonce=JDE0SYJCAAqWUwWJiBM=; SelectiveEncryption=1
**a=control:rtsp://example.com/session0000012838984/m1**
a=3GPP-Adaptation-Support:2
**a=3GPP-SRTP-Config:3NivNiiwMNgZmngs128OcA== NRknve/o/LXY97cRY7Y= auth-tag-len=32**
**m=video 0 RTP/SAVP 99 100**
b=AS:64
b=RR:2000
b=RS:1200
a=rtpmap:99 H263-2000/90000
a=fmtp:99 profile=3;level=10
a=rtpmap:100 RTP.ENC.AESCM128/90000
a=fmtp:100 opt=99; ContentID="content6188164@ContentIssuer.com"; RightsIssuerURL="
http://drm.rightsserver.org/6188164"; IVnonce= IwOSRWeSAUiVEiN5gVA=
**a=control:rtsp://example.com/session0000012838984/m2**
a=3GPP-Adaptation-Support:1
**a=3GPP-SRTP-Config:PyChokXYVigC9kDftofE7Q== 0zvrjkBK/9Yc3BJ61/Q= auth-tag-len=80**

This SDP is then transmitted to the client.

5. The client decrypts k, derives the keys Ks and Km, and verifies the integrity of the SDP part. The freshness token's validity needs also to be checked. If successful, the clients populates the SRTP crypto contexts using the supplied keys and parameters. The client uses RTSP to setup both media streams in an aggregated session at server. This is done using the new control URI supplied in the SDP, which allows the server to determine which of its generated contexts shall be used for this session.

6. The client requests to start media deliver through a RTSP PLAY request. The server responds.

7. The server delivers a stream of SRTP packets that are integrity protected (as well as pre-encrypted, in accordance to section K.1).

# Annex XK (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 |
| 06-2004 | 24 | SP-040434 | 068 | 1 | Addition of Release-6 functionality | 5.6.0 | 6.0.0 |