

INTERNATIONAL STANDARD



**Consumer terminal function for access to IPTV and open internet multimedia services –
Part 4-1: Protocols**

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2017 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IECNORM.COM : Click to view the full text of IEC 60366-4-1:2017

INTERNATIONAL STANDARD



**Consumer terminal function for access to IPTV and open internet multimedia services –
Part 4-1: Protocols**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 33.170 35.240.95

ISBN 978-2-8322-4615-3

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	14
INTRODUCTION.....	16
1 Scope.....	17
2 Normative references	17
3 Terms, definitions and abbreviations	22
3.1 Terms and definitions.....	22
3.2 Abbreviated terms.....	23
4 Interface definitions	23
4.1 Consumer network to provider network interfaces (UNI).....	23
4.2 Provider network reference points description	25
4.3 Interfaces to external systems, consumer network	28
5 Structure of this document.....	28
6 HTTP.....	29
6.1 HTTP reference points	29
6.2 IG as a protocol converter supporting the HNI-IGI interface – HTTP option	30
6.3 Protocols for IPTV service functions.....	30
6.3.1 Multicast content streaming with SIP session management protocol over HNI-IGI – HTTP option.....	30
6.3.2 Unicast content streaming with SIP session management.....	42
6.3.3 Forced play-out control with SIP session management – Protocol for forced play-out control over HNI-IGI	50
6.3.4 Content download.....	50
6.3.5 Purchase of digital media service using SIP	50
6.3.6 Pay per view multicast content service with SIP session management protocol over HNI-IGI – HTTP option	57
6.3.7 Parental control for content using SIP.....	58
6.3.8 Network-based user notification services	66
6.3.9 Content bookmarking.....	69
6.3.10 Local PVR service using SIP	77
6.3.11 Network PVR (nPVR) using SIP.....	82
6.3.12 Personalised channel	84
6.3.13 Session transfer with SIP session management.....	85
6.4 Protocol for service access and control functions.....	92
6.4.1 Service provider discovery.....	92
6.4.2 Service discovery	97
6.4.3 Service access	97
6.4.4 Subscription profile management and usage.....	98
6.4.5 Remote management.....	103
6.4.6 User registration and network authentication	108
6.5 Protocols for communications functions using SIP	119
6.5.1 CallerID	119
6.5.2 Instant messaging Procedure for instant messaging on HGI-IGI – HTTP option	121
6.5.3 IM session (chat using MSRP).....	123
6.5.4 Presence	134
6.5.5 Content sharing	139
6.6 Protocols system infrastructure functions OITF-IG interface (HNI-IGI).....	149

6.6.1	HNI-IGI message types.....	149
6.6.2	HNI-IGI messages in the OITF to IG direction – HTTP option	150
6.6.3	HNI-IGI messages in the IG to OITF direction – HTTP option	152
6.6.4	HNI-IGI PENDING_IG message – HTTP option.....	153
6.6.5	HNI-IGI SIP message – HTTP option	154
6.6.6	HNI-IGI auxiliary message	154
6.6.7	HNI-IGI message body – HTTP option	155
6.6.8	Guidelines for applications using the HNI-IGI interface – HTTP option.....	155
6.6.9	Error recovery in the IG – HTTP option	155
6.7	Protocols for content preparation	156
6.7.1	Reference points NPI-45, NPI-46, NPI-CSPT3 and NPI-CSPG3.....	156
6.7.2	XML schema for content preparation	176
7	SIP and SIP/SDP	181
7.1	General.....	181
7.2	SIP/SDP reference points within the provider network.....	182
7.2.1	General	182
7.2.2	Generic handling for SIP requests/responses	182
7.2.3	Protocols for IPTV service functions	182
7.2.4	Protocols for service access and control functions.....	204
7.2.5	Protocols for communication services using SIP CallerID Procedures for caller ID on UNIS-8.....	206
7.3	SIP/SDP reference points within the residential network	208
7.3.1	General	208
7.3.2	IG as a B2BUA	208
7.3.3	Protocols for IPTV service functions	209
7.3.4	Protocols for service access and control functions.....	220
7.3.5	Protocols for communication services using SIP	224
8	RTSP	235
8.1	General.....	235
8.2	Protocols for IPTV service functions.....	236
8.2.1	Use of RTSP for unicast content streaming services.....	236
8.2.2	Use of RTSP for forced play-out control RTSP for managed model over UNIS-11 and NPI-10	243
8.2.3	Use of RTSP for network PVR (nPVR)	244
8.2.4	Use of RTSP for personalized channel	244
8.2.5	Content streaming post-session transfer.....	245
8.3	Protocols for service access and control Performance monitoring over UNIT-18	245
9	IGMP and multicast protocol.....	248
9.1	General.....	248
9.2	Protocols for IPTV service functions.....	249
9.2.1	Multicast content streaming service on UNIS-13	249
9.2.2	Pay-per-view multicast content service with SIP session management	249
9.2.3	nPVR Protocol over NPI-40	249
9.2.4	Network-generated notification service	249
9.2.5	Emergency notification service	250
9.3	Protocols for service access and control functions	250
9.3.1	Service discovery and Content Selection	250
9.3.2	Remote management – protocol over UNI-RMS	250

9.4	Protocols for system infrastructure functions – interactive application delivery – protocol over UNIS-6 and UNIS-12	250
10	RTP/RTCP.....	251
10.1	General.....	251
10.2	Protocols for IPTV service functions.....	251
10.2.1	Multicast content streaming service	251
10.2.2	Unicast content streaming service	251
10.2.3	nPVR Protocol over NPI-41	251
10.3	Service access and control – Performance monitoring over UNIT-18	251
10.4	Application-layer forward error correction.....	252
10.5	Application-layer retransmission (RET)	252
10.5.1	General	252
10.5.2	Protocol over UNIT-17	252
10.5.3	Protocol over UNIT-18	253
10.6	Fast channel change (FCC)	253
10.6.1	General	253
10.6.2	Protocol over UNIT-17	253
10.6.3	Protocol over UNIT-18	253
10.7	Protocols for multimedia telephony services – protocol over UNIT-17	253
10.7.1	General	253
10.7.2	RTP payload formats for video streams.....	253
10.7.3	RTP payload formats for audio streams.....	253
11	UPnP protocols for system infrastructure functions – UPnP discovery	254
11.1	General.....	254
11.2	Procedure for IG discovery	254
11.2.1	Discovery sequence	254
11.2.2	urn:oipf-org:device:ig:1 device definitions	254
11.2.3	IG description	254
11.3	Procedure for AG discovery	257
11.3.1	Discovery sequence	257
11.3.2	urn:oipf-org:device:ag:1 device definitions	257
11.3.3	AG description.....	257
11.4	Procedure for CSPG-DTCP discovery	258
11.4.1	Discovery sequence	258
11.4.2	urn:oipf-org:device:cspg-dtcp:1 device definitions.....	258
11.4.3	CSPG-DTCP description.....	259
11.5	Procedure for OITF discovery	260
11.5.1	Discovery sequence	260
11.5.2	OITF description.....	260
12	DLNA function	260
13	DHCP	260
13.1	General.....	260
13.2	Protocols for system infrastructure functions – network attachment.....	261
13.2.1	General	261
13.2.2	DHCP option usage	261
14	UDP for IPTV service functions.....	263
14.1	Multicast content streaming service – protocol over UNIT-17	263
14.2	Unicast content streaming service – Protocol over UNIT-17	263

14.3	nPVR – protocol over NPI-41	263
15	FLUTE.....	263
15.1	General.....	263
15.2	Protocols for IPTV service Functions	264
15.2.1	Emergency notification	264
15.2.2	Network-generated notification	264
16	Diameter.....	264
16.1	General.....	264
16.2	Protocols for IPTV service functions – Purchase of digital media – Protocol over NPI-11	264
Annex A	(informative) Example messages	265
A.1	IPTV service functions message examples – Unicast content streaming session setup with SIP session management	265
A.2	Communication services message examples – HNI-IGI message mapping to SIP	269
A.2.1	General	269
A.2.2	Presence	270
A.2.3	Chat	275
A.2.4	Presence document.....	279
Annex B	(informative) User profile description	280
B.1	IPTV subscription profile	280
B.1.1	General	280
B.1.2	XML Schema for the IPTV subscription profile	280
B.2	XML schema for the OIPF profile	285
B.3	IPTV subscription profile elements classification	290
B.3.1	General	290
B.3.2	User visible and manageable data	290
B.3.3	User visible, but not manageable data	290
B.3.4	Data neither visible nor manageable by the user.....	291
Annex C	(normative) Mapping attributes for scheduled content	292
C.1	Mapping SDP attributes from DVB SD&S information	292
C.2	Service package SDP attributes	294
Annex D	(normative) <protocol> names	295
Annex E	(normative) System infrastructure	296
E.1	OITF start-up high-level procedures	296
E.1.1	OITF with native HNI-IGI support	296
E.1.2	OITF with non-native HNI-IGI support	297
E.1.3	Integrated OITF/IG with no HNI-IGI support	298
E.2	High-level procedure for an OITF graceful shut-down in network relying on IMS.....	299
E.3	OITF restart high-level procedures for an IG integrating WAN GW	300
E.4	IG start-up and shutdown procedures	302
E.4.1	IG start-up procedures	302
E.4.2	IG shutdown procedures	302
E.5	WAN Gateway functions	303
E.5.1	General	303
E.5.2	NAT traversal	303
E.5.3	NAT traversal for SIP-based services	303

E.5.4	NAT traversal and keep-alive messages for unicast content streaming services	305
Annex F (informative)	System infrastructure mechanisms – NAT-T informational flows for IPTV services with SIP session management	307
F.1	General.....	307
F.2	IG and WAN GW in one physical device.....	307
F.3	IG and WAN GW in different physical devices	309
Annex G (informative)	NAT-T informational flows for IPTV services	312
Annex H (normative)	Port mapping and NAT traversal for FCC/RET for multicast content services	313
H.1	General.....	313
H.2	Cookie signalling	313
H.3	NAT traversal without cookie signalling	315
H.4	Correlation of RTCP messages sent to the RS and FT entity	316
Annex I (normative)	Presence XML schema.....	317
Annex J (informative)	Protocol procedure subclause structure	318
Annex K (normative)	OITF-specific TR-135 and TR-106 remote management objects	320
K.1	General.....	320
K.2	OITF-specific TR-135 remote management object.....	320
K.3	OITF-specific TR-106 remote management object.....	326
Annex L (informative)	New event package for SIP SUBSCRIBE/NOTIFY	329
Annex M (informative)	Overview of notification services in OIPF R2.....	330
Annex N (normative)	Fast Channel Change and Retransmission (FCC/RET).....	332
N.1	Application-layer retransmission (RET)	332
N.1.1	General	332
N.1.2	Unicast RET for multicast content service.....	332
N.1.3	Unicast RET for unicast content service.....	333
N.1.4	Multicast RET for multicast content service.....	334
N.1.5	Packet loss event upstream of the RET server.....	334
N.1.6	Packet loss event downstream of the RET server	335
N.2	Fast channel change (FCC)	336
Annex O (informative)	IG handling of IMPUs in association with GRUU	339
Annex P (normative)	FDT schema extensions	340
Annex Q (normative)	IG Service awareness	341
Q.1	General.....	341
Q.2	IG Algorithm for service awareness.....	341
Annex R (normative)	Definition of content reporting info package	342
R.1	General.....	342
R.2	Applicability	342
R.3	Info Package Name.....	342
R.4	Info Package Parameters	342
R.5	SIP option Tags	343
R.6	INFO message body parts.....	343
R.6.1	General	343
R.6.2	SIP Content-Type header field value	343
R.6.3	SIP Content-Disposition header field value	343
R.6.4	Message body syntax	343
R.7	Info package usage restrictions.....	343

R.8	Rate of INFO requests	343
R.9	Info Package Security Considerations	343
Annex S	(normative) Definition of Digital-Media-Purchase info package.....	344
S.1	General.....	344
S.2	Applicability	344
S.3	Info package name	344
S.4	Info Package parameters	344
S.5	SIP option Tags	344
S.6	INFO message body parts.....	345
S.6.1	General	345
S.6.2	SIP Content-Type header field value	345
S.6.3	SIP Content-Disposition header field value	345
S.6.4	Message body syntax	345
S.6.5	Info Package usage restrictions	345
S.7	Rate of INFO requests	345
S.8	Info package security considerations	345
Annex T	(normative) Definition of parental control watched content info package.....	346
T.1	General.....	346
T.2	Applicability	346
T.3	Info package name	346
T.4	Info package parameters	346
T.5	SIP option tags	346
T.6	INFO message body parts.....	346
T.6.1	General	346
T.6.2	SIP Content-Type header field value	347
T.6.3	SIP Content-Disposition header field value	347
T.6.4	Message body syntax	347
T.7	Info package usage restrictions.....	347
T.8	Rate of INFO requests	347
T.9	Info package security considerations	347
Annex U	(normative) Common types	348
U.1	Schema	348
U.2	Definitions.....	348
U.2.1	ProgramIdType	348
U.2.2	UserIdType	348
Annex V	(normative) Schema extension for FLUTE FDT	349
V.1	Namespace.....	349
V.2	Import Namespace and schema	349
V.3	Extension of FDT attributes.....	349
Bibliography		350
Figure 1	– Elements of OIPF extended RelatedMaterialType	51
Figure 2	– Construction of advertising window using (x, y, w, h) in Position_Size element.....	51
Figure 3	– Overlaying the content with advertisement for related media	52
Figure 4	– Sequence for the configuration of an IG	105
Figure 5	– GetKeyScheduleSignalingrequest schema	158
Figure 6	– GetKeyScheduleSignalingresponse schema.....	160

Figure 7 – SetScheduleAndGetKeySignalingrequest schema	163
Figure 8 – SetScheduleAndGetKeySignalingresponse schema	164
Figure 9 – SetKeyScheduleAndGetSignalingrequest schema	167
Figure 10 – SetKeyScheduleAndGetSignalingresponse schema	168
Figure 11 – IG description XML schema.....	255
Figure A.1 – COD session setup sequence	265
Figure E.1 – High-level start-up procedural flow for an OITF with native HNI-IGI support	297
Figure E.2 – High-level start-up procedural flow for an OITF without native HNI-IGI support	298
Figure E.3 – High-level start-up procedural flow for an integrated OITF/IG.....	299
Figure E.4 – High-level shut-down procedural flow for an OITF	300
Figure E.5 – Overview of OITF start-up.....	301
Figure E.6 – Overview of OITF restart.....	302
Figure E.7 – NAT traversal with UDP keep alive messages.....	304
Figure E.8 – NAT traversal with SIP keep alive messages	305
Figure F.1 – Protocol flows for IG and WAN GW in one device	307
Figure F.2 – Simplified unicast content streaming service information flows.....	308
Figure F.3 – NAT traversal based on IMS access NAT-T model	309
Figure F.4 – Workflow for IMS NAT-T model.....	310
Figure G.1 – NAT traversal based on the Symmetric-RTP mechanism	312
Figure H.1 – Port mapping process and keep-alive packets for RET	314
Figure H.2 – NAPT traversal for FCC/RET with cookie signalling	314
Figure H.3 – NAPT traversal for FCC/RET without cookie signalling	315
Figure J.1 – Subclause structure in the present annex	318
Figure J.2 – TISPAN IMS IPTV Stage 3 specification structure	319
Figure M.1 – Overview of notification services	330
Figure N.1 – Call flow for multicast content service with unicast retransmission service	333
Figure N.2 – Example showing location of packet loss deep in the network relative to the RET server	334
Figure N.3 – Multicast RET protocol interaction for packet loss upstream of RET server	335
Figure N.4 – Multicast RET protocol interaction for packet loss downstream of RET server, with the OITF acting as immediate reporter	336
Figure N.5 – Multicast content delivery with FCC	338
Table 1 – UNI reference points and protocols	24
Table 2 – Other interfaces	25
Table 3 – NPI reference points and protocols.....	26
Table 4 – External interfaces from the consumer network	28
Table 5 – Supported HTTP extension headers in the HNI-IGI INVITE request message for multicast content service session setup (OITF→IG).....	33
Table 6 – Supported HTTP extension headers in the response message to an HNI-IGI INVITE request message for multicast content service session setup (IG→OITF)	34
Table 7 – Supported HTTP extension headers in the HNI-IGI ACK message for a successful multicast content service session setup (OITF→IG).....	34

Table 8 – Supported HTTP extension headers in HNI-IGI BYE request for teardown of a multicast content streaming session (OITF→IG)	36
Table 9 – Supported HTTP extension headers in the response to an HNI-IGI BYE request for teardown of a multicast content streaming session (IG→OITF).....	36
Table 10 – Supported HTTP extension headers in the HNI-IGI SIP INFO request message for content reporting (OITF→IG)	38
Table 11 – Supported HTTP extension headers in the response message to an HNI-IGI INFO request message for content reporting (IG→OITF)	38
Table 12 – List of HTTP extension headers for an incoming SIP UPDATE (IG→OITF)	39
Table 13 – Supported HTTP extension headers in the response message to an incoming HNI-IGI SIP UPDATE request message (OITF→IG)	39
Table 14 – Supported HTTP extension headers in HNI-IGI OPTION request for unicast content streaming session setup parameters (OITF→IG)	43
Table 15 – Supported HTTP extension headers in the response to an HNI-IGI OPTION request for unicast content streaming session setup parameters.....	43
Table 16 – Supported HTTP extension headers in HNI-IGI INVITE request for unicast content streaming session setup (OITF→IG).....	47
Table 17 – Supported HTTP extension headers in the response to an HNI-IGI INVITE request for unicast content streaming session setup (IG→OITF).....	48
Table 18 – Supported HTTP extension headers in HNI-IGI ACK request for successful unicast content streaming session teardown (OITF→IG)	48
Table 19 – Supported HTTP extension headers in HNI-IGI BYE request for unicast content streaming session teardown (OITF→IG)	49
Table 20 – Supported HTTP extension headers in the response to an HNI-IGI BYE request for unicast content streaming session teardown (IG→OITF)	49
Table 21 – Supported HTTP extension headers in the HNI-IGI SIP INFO request message for Digital-Media-Purchase (OITF→IG)	53
Table 22 – Supported HTTP extension headers in the response message to an HNI-IGI INFO request message for Digital-Media-Purchase (IG→OITF)	53
Table 23 – List of HTTP extension headers for an incoming SIP UPDATE (IG→OITF)	54
Table 24 – Supported HTTP extension headers in the response message to an incoming HNI-IGI SIP UPDATE request message (OITF→IG)	54
Table 25 – Mandatory HTTP headers in the request message for advertisement of digital media (OITF→IPTV Metadata control FE).....	55
Table 26 – Mandatory HTTP headers in the response message for advertisement of digital media (IPTV Metadata control FE→OITF).....	55
Table 27 – Supported HTTP extension headers in HNI-IGI SUBSCRIBE request for the parental control watched content event package	59
Table 28 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request for the parental control watched content event package.....	60
Table 29 – Supported HTTP extension headers in the NOTIFY request containing changes in the watched programme (IG→OITF).....	60
Table 30 – Supported HTTP extension headers in the response to a NOTIFY request (OITF→IG).....	60
Table 31 – Supported HTTP extension headers in the MESSAGE request for parental control (OITF→IG)	63
Table 32 – Supported HTTP extension headers in the response message to a MESSAGE request message for parental control (IG→OITF)	63
Table 33 – Supported HTTP extension headers in the MESSAGE request for parental control (IG→OITF)	64

Table 34 – List of HTTP extension headers for an outgoing SIP MESSAGE for setting up a user notification request (OITF→IG).....	67
Table 35 – List of HTTP extension headers for the response to an outgoing SIP MESSAGE for setting up a user notification request (IG→OITF)	67
Table 36 – Supported HTTP extension headers for an incoming SIP INFO (IG→OITF)	71
Table 37 – Supported HTTP extension headers in the response message to an incoming HNI-IGI SIP INFO (OITF→IG)	72
Table 38 – Supported HTTP extension headers in the HNI-IGI SIP INFO request message for content bookmark (OITF→IG)	74
Table 39 – Supported HTTP extension headers in the response message to an HNI-IGI INFO request message for content bookmark (IG→OITF).....	74
Table 40 – List of HTTP extension headers for an incoming SIP UPDATE (IG→OITF).....	75
Table 41 – Supported HTTP extension headers in the response message to an incoming HNI-IGI SIP UPDATE request message (OITF→IG)	75
Table 42 – List of HTTP extension headers for PVR service capture request message (OITF→IG).....	79
Table 43 – List of HTTP extension headers for the response to PVR service capture request and PVR record request message (IG→OITF and OITF→IG)	79
Table 44 – List of HTTP extension headers for the PVR record request (IG→OITF)	80
Table 45 – List of HTTP extension headers for the PVR record request (IG→OITF)	83
Table 46 –HTTP extension headers for SIP REFER.....	89
Table 47 –HTTP extension headers SIP REFER response.....	90
Table 48 –HTTP extension headers in the NOTIFY request	91
Table 49 –HTTP extension headers in the NOTIFY response.....	91
Table 50 – Supported HTTP extension headers in HNI-IGI SUBSCRIBE request for SP discovery	95
Table 51 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request for SP discovery	96
Table 52 – Supported HTTP extension headers in the NOTIFY request to the SUBSCRIBE to SP discovery.....	96
Table 53 – Supported HTTP extension headers in the response to a NOTIFY request to the SUBSCRIBE to SP discovery	96
Table 54 – Supported HTTP extension headers in HNI-IGI SUBSCRIBE request for receiving notification of changes in the IPTV service profile	101
Table 55 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request.....	101
Table 56 – Supported HTTP extension headers in the NOTIFY request containing changes in the IPTV service profile	102
Table 57 – Supported HTTP extension headers in the response to a NOTIFY request	102
Table 58 – RPC methods	103
Table 59 – RPC method parameters	104
Table 60 –Mandatory HTTP extension headers for user registration/de-registration (OITF→IG).....	111
Table 61 –HTTP extension headers for user registration/de-registration response (IG→OITF).....	112
Table 62 – Supported HTTP extension headers in the HNI-IGI SUBSCRIBE request for the registration event package	113
Table 63 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request for the registration event package.....	114

Table 64 –HTTP extension headers for a HNI-IGI NOTIFY request sent IG→OITF	114
Table 65 –HTTP extension headers in the response to a NOTIFY request	114
Table 66 –HTTP extension headers for an instant message based caller ID (IG→OITF)	119
Table 67 –HTTP extension headers for the response to an Instant message based caller ID (OITF→IG).....	120
Table 68 –HTTP extension headers on the HNI-IGI interface (IG→OITF) for a received SIP INVITE	121
Table 69 –HTTP extension headers on the HNI-IGI interface (OITF→IG) for a response to the SIP INVITE	121
Table 70 –HTTP headers in the HNI-IGI ACK message (IG→OITF).....	121
Table 71 –HTTP extension headers for an outgoing instant message (OITF→IG)	122
Table 72 –HTTP extension headers for the response to an outgoing and incoming Instant message (IG→OITF and OITF→IG).....	122
Table 73 – List of HTTP extension headers for an incoming instant message (IG→OITF).....	123
Table 74 –HTTP extension headers for IM INVITE request (OITF→IG)	125
Table 75 –HTTP extension headers for a 200 OK response received for the INVITE IG→OITF	125
Table 76 – List of HTTP extension headers in HNI-IGI ACK request	126
Table 77 –HNI-IGI HTTP extension headers for an MSRP SEND request (OITF→IG)	127
Table 78 –HNI-IGI HTTP extension headers included in the HTTP 200 OK response (IG→OITF).....	127
Table 79 –HNI-IGI HTTP extension headers for an MSRP SEND ACTIVITY request (OITF→IG).....	128
Table 80 –HNI-IGI HTTP extension headers for an incoming MSRP message (IG→OITF).....	128
Table 81 –HNI-IGI HTTP extension headers for an MSRP 200 OK response to an incoming MSRP message (OITF→IG)	128
Table 82 –HNI-IGI HTTP extension headers for an incoming MSRP receive activity (IG→OITF).....	129
Table 83 –HTTP extension headers for an MSRP BYE request (OITF→IG).....	130
Table 84 –HTTP extension headers for a 200 OK response to a BYE (IG→OITF)	130
Table 85 –HTTP extension headers for an incoming SIP BYE (IG→OITF).....	131
Table 86 –HTTP extension headers for the response to an SIP BYE (OITF→IG).....	131
Table 87 –HTTP extension headers for an incoming IM INVITE request (IG→OITF).....	133
Table 88 – List of HTTP extension headers for the response to an incoming IM INVITE request (OITF→IG)	133
Table 89 – Supported HTTP extension headers in HNI-IGI ACK request for successful IM session (MSRP chat) (IG→OITF)	134
Table 90 –HTTP extension headers for a SUBSCRIBE request (OITF→IG).....	136
Table 91 –HTTP extension headers for the response to a SUBSCRIBE to presence (IG→OITF).....	136
Table 92 –HTTP extension headers for a SIP NOTIFY (IG→OITF).....	137
Table 93 – List of HTTP extension headers for a response to a received SIP NOTIFY (OITF→IG).....	137
Table 94 –HTTP extension headers for the PUBLISH request (OITF→IG).....	138
Table 95 –HTTP extension headers for a response to SIP PUBLISH (IG→OITF).....	139

Table 96 –HTTP extension headers on the HNI-IGI interface (OITF→IG) for a SIP OPTIONS request and response (IG→OITF).....	141
Table 97 –HTTP extension headers on the HNI-IGI interface (OITF→IG) for a SIP INVITE request	142
Table 98 –HTTP extension headers for a 200 OK response received for the INVITE IG→OITF	143
Table 99 –HTTP extension headers in HNI-IGI ACK request	143
Table 100 –HTTP extension headers for SIP REFER	145
Table 101 – Supported HTTP extension headers in the NOTIFY request sent from the transferee OITF (OITF->IG) to the transferor (IG->OITF)	146
Table 102 –HTTP extension headers for a SIP BYE request (OITF→IG)	148
Table 103 –HTTP extension headers for a 200 OK response to a BYE (IG→OITF)	149
Table 104 – HNI-IGI message types	150
Table 105 – X-OITF HTTP extension headers and IG actions for OITF→IG messages	151
Table 106 – Mapping of SIP header to X-OITF HTTP extension headers in IG→OITF	152
Table 107 – Reference points calling entity and called entity	157
Table 108 – Operations according to functional entity roles	157
Table 109 – getKeyScheduleSignaling method parameters	158
Table 110 – GetKeyScheduleSignalingrequestType	159
Table 111 – GetKeyScheduleSignalingresponseType	161
Table 112 – setScheduleAndGetKeySignaling method parameters.....	162
Table 113 – SetScheduleAndGetKeySignalingrequestType	163
Table 114 – SetScheduleAndGetKeySignalingresponseType	165
Table 115 – setKeyScheduleAndGetSignaling method parameters.....	166
Table 116 – SetKeyScheduleAndGetSignalingrequestType	167
Table 117 – SetKeyScheduleAndGetSignalingresponseType	169
Table 118 – UUIDType	170
Table 119 – StatusType	170
Table 120 – ContentKeyType structure	170
Table 121 – DrmInfoType structure	171
Table 122 – DrmMetadataType structure	171
Table 123 – DrmSignalingType structure	171
Table 124 – DashSignalingType structure	172
Table 125 – EncryptionprofileType structure	172
Table 126 – KeyIdScheduleType structure	173
Table 127 – PregenerationWindowType structure	173
Table 128 – PrivateSignalingType structure	173
Table 129 – ScheduledKeyType structure	174
Table 130 – ScheduleType structure	175
Table 131 – TimeReferenceType structure.....	175
Table 132 –SIP headers for PVR service recording status (CC→IPTV Control)	198
Table 133 –SIP headers for PVR service recording status (CC→IPTV Control)	200
Table 134 – Supported SIP headers in the HNI-IGI INVITE request message for an outgoing MM telephony session setup (OITF→IG)	231

Table 135 – Supported SIP headers in the response message to an HNI-IGI INVITE request message for an outgoing MM telephony session setup (IG→OITF)	231
Table 136 – Supported SIP headers in the HNI-IGI ACK message for a successful MM telephony session setup (OITF→IG)	232
Table 137 – Supported SIP headers in the HNI-IGI INVITE request message for an incoming MM telephony session setup (IG→OITF)	232
Table 138 – Supported SIP headers in the response message to an HNI-IGI INVITE request message for an incoming MM telephony session setup (OITF→IG)	233
Table 139 – Supported SIP headers in the HNI-IGI ACK message for a successful incoming MM telephony session setup (IG→OITF)	233
Table 140 – List of SIP headers for an outgoing SIP BYE (OITF→IG)	234
Table 141 – List of SIP headers for the response to an outgoing SIP BYE (IG→OITF)	234
Table 142 – List of SIP headers for an incoming SIP BYE (IG→OITF)	234
Table 143 – List of SIP headers for the response to an incoming SIP BYE (OITF→IG)	235
Table 144 – urn:oipf-org:device:ig:1 deviceType	254
Table 145 – igDescription elements and attributes	256
Table 146 – urn:oipf-org:device:ag:1 deviceType	257
Table 147 – agDescription elements and attributes	258
Table 148 – urn:oipf-org:device:cspg-dtcp:1 deviceType	258
Table 149 – cspgdtcpDescription elements and attributes	259
Table 150 – urn:oipf-org:device:oif:1 deviceType	260
Table C.1 – Definition of <protocol> names	292
Table D.1 – Definition of <protocol> names	295
Table K.1 – Parameter list for an OITF using TR-135	320
Table K.2 – Parameter list for an OITF using TR-106	327
Table M.1 – Summary of notification mechanisms	331

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**CONSUMER TERMINAL FUNCTION FOR ACCESS TO IPTV AND
OPEN INTERNET MULTIMEDIA SERVICES –****Part 4-1: Protocols****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62766-4-1 has been prepared by IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this International Standard is based on the following documents:

CDV	Report on voting
100/2546/CDV	100/2660/RVC

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62766 series, published under the general title *Consumer terminal function for access to IPTV and open internet multimedia services*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

INTRODUCTION

The IEC 62766 series is based on a series of specifications that was originally developed by the OPEN IPTV FORUM (OIPF). They specify the user-to-network interface (UNI) for consumer terminals to access IPTV and open internet multimedia services over managed or non-managed networks as defined by OIPF.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

CONSUMER TERMINAL FUNCTION FOR ACCESS TO IPTV AND OPEN INTERNET MULTIMEDIA SERVICES –

Part 4-1: Protocols

1 Scope

This part of IEC 62766 specifies the protocols, which apply to the following reference point interfaces identified in the architecture described in Annex B of IEC 62766-1:2017.

- The UNI interfaces, between the consumer domain and the network or service provider domains.
- The HNI interfaces, between the functional entities in the consumer network domain.
- Interfaces to external systems, which include DLNA networks in the consumer domain.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62481-2:2013, *Digital living network alliance (DLNA) home networked device interoperability guidelines – Part 2: Media Formats*

IEC 62766-1, *Consumer terminal function for access to IPTV and open Internet multimedia services – Part 1: General*

IEC 62766-2-1, *Consumer terminal function for access to IPTV and open Internet multimedia services – Part 2-1: Media formats*

IEC 62766-3, *Consumer terminal function for access to IPTV and open Internet multimedia services – Part 3: Content metadata*

IEC 62766-4-2, *Consumer terminal function for access to IPTV and open Internet multimedia services – Part 4-2: Examples of IPTV protocol sequences*

IEC 62766-5-1, *Consumer terminal function for access to IPTV and open Internet multimedia services – Part 5-1: Declarative application environment*

IEC 62766-7, *Consumer terminal function for access to IPTV and open Internet multimedia services – Part 7: authentication, content protection and service protection*

ISO/IEC 13818-1:2015, *Information technology – Generic coding of moving pictures and associated audio information – Part 1: Systems*

ISO/IEC 14496-3:2009, *Information technology – Coding of audio-visual objects – Part 3: Audio*

ISO/IEC 14496-10:2005, *Information technology – Coding of audio-visual objects – Part 10: Advanced video coding*

ISO/IEC 29341-4-13, *Information technology – UPnP device architecture – Part 4-13: Audio video device control protocol –Level 2 – Rendering control service*

ETSI ES 282 003, *Resource and Admission Control Sub-system (RACS) – functional architecture*

ETSI ES 282 010, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); Charging*

ETSI ES 283 002, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); H.248 profile for controlling Access and Residential Gateways*

ETSI ES 283 003, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); IP Multimedia Call Control based on session Initiation Protocol (SIP) and session Description Protocol (SDP) Stage 3 [3GPP TS 24.229 [Release 7], modified]*

ETSI ES 283 034, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); Network Attachment Sub-System (NASS); e4 interface based on the DIAMETER protocol*

ETSI TS 102 034 V1.5.1, *Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB services over IP Based Networks*

ETSI TS 102 539, *Digital Video Broadcasting (DVB); Carriage of Broadband Content Guide (BCG) information over Internet Protocol (IP)*

ETSI TS 102 822-3-1, *Broadcast and On-line services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 3: Metadata; Sub-part 1: Metadata schemas*

ETSI TS 103 197 V1.5.1, *Digital Video Broadcasting (DVB); Head-end implementation of DVB SimulCrypt*

ETSI TS 124 237, *Universal Mobile Telecommunications System (UMTS); LTE; IP Multimedia (IM) Core Network (CN) subsystem IP Multimedia Subsystem (IMS) service continuity; Stage 3*

ETSI TS 124 503, 3GPP TS 24.229 (Release 7) modified, 3GPP TS 24.503 (Release 8), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); TISPAN; IP Multimedia Call Control Protocol based on session Initiation Protocol (SIP) and session Description Protocol (SDP) Stage 3*

ETSI TS 126 237, *Universal Mobile Telecommunications System (UMTS); LTE; IP Multimedia Subsystem (IMS) based Packet Switch Streaming (PSS) and Multimedia Broadcast/Multicast Service (MBMS) User Service; Protocols*

ETSI TS 129 329, 3GPP TS 29.329 V7.4.0 Release 7, *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Sh interface based on the Diameter protocol; Protocol details*

ETSI TS 181 005, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); service and Capability Requirements*

ETSI TS 183 017, *Telecommunications and Internet Converged services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control: DIAMETER protocol for*

session based policy set-up information exchange between the application Function (AF) and the service Policy Decision Function (SPDF); Protocol specification

ETSI TS 183 019, *Network Attachment: User-Network protocol Interface Definitions*

ETSI TS 183 033, 3GPP TS 29.228 V6.8.0 and 3GPP TS 29.229 V6.6.0 modified, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); IP Multimedia; Diameter based protocol for the interfaces between the Call session control function and the User profile Server Function/Subscription Locator Function; Signalling flows and protocol details*

ETSI TS 183 060, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control Subsystem (RACS); Re interface based on the DIAMETER protocol*

ETSI TS 183 063, *Telecommunications and Internet converged services and Protocols for Advanced Networking (TISPAN); IMS-based IPTV stage 3 specification, Release 3*

IETF RFC 1918, *Address Allocation for Private Internets*

IETF RFC 2045, *Multipurpose Internet Mail extensions (MIME) Part One: Format of Internet message Bodies*

IETF RFC 2132, *DHCP options and BOOTP Vendor extensions*

IETF RFC 2326, *Real Time Streaming Protocol (RTSP)*

IETF RFC 2387, *The MIME Multipart/Related Content-type*

IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*

IETF RFC 2617, *HTTP authentication: Basic and Digest Access authentication*

IETF RFC 3016, *RTP Payload Format for MPEG-4 Audio/Visual Streams*

IETF RFC 3261, *SIP: session Initiation Protocol*

IETF RFC 3264, *An Offer/Answer Model with the session Description Protocol (SDP)*

IETF RFC 3265, *session Initiation Protocol (SIP)-Specific Event Notification*

IETF RFC 3376, *Internet Group Management Protocol, Version 3*

IETF RFC 3420, *Internet Media Type message/sipfrag*

IETF RFC 3428, *session Initiation Protocol (SIP) extension for Instant Messaging*

IETF RFC 3442, *The Classless Static Route option for Dynamic Host Configuration Protocol (DHCP) version 4*

IETF RFC 3515, *The session Initiation Protocol (SIP) REFER Method*

IETF RFC 3550, *RTP: A Transport Protocol for Real-Time applications*

IETF RFC 3551, *RTP profile for Audio and Video Conferences with Minimal Control*

IETF RFC 3556, *session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth*

IETF RFC 3588, *Diameter Base Protocol*

IETF RFC 3611, *RTP Control Protocol Extended Reports (RTCP XR)*

IETF RFC 3621, *SIP: Session Initiation Protocol*

IETF RFC 3640, *RTP Payload Format for Transport of MPEG-4 Elementary Streams*

IETF RFC 3680, *A session Initiation Protocol (SIP) Event Package for Registrations*

IETF RFC 3840, *Indicating User Agent Capabilities in the session Initiation Protocol (SIP)*

IETF RFC 3841, *Caller Preferences for the session Initiation Protocol (SIP)*

IETF RFC 3856, *A Presence Event Package for the session Initiation Protocol (SIP)*

IETF RFC 3890, *A Transport Independent Bandwidth Modifier for the session Description Protocol (SDP)*

IETF RFC 3891, *The session Initiation Protocol (SIP) "Replaces" header*

IETF RFC 3903, *session Initiation Protocol (SIP) extension for Event State Publication*

IETF RFC 3925, *Vendor-Identifying Vendor options for Dynamic Host Configuration Protocol version 4 (DHCPv4)*

IETF RFC 3926, *FLUTE – File Delivery over Unidirectional Transport*

IETF RFC 3984, *RTP Payload Format for H.264 Video*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*

IETF RFC 3994, *Indication of message Composition for Instant Messaging*

IETF RFC 4028, *Session Timers in the Session Initiation Protocol (SIP)*

IETF RFC 4122, *A Universally Unique Identifier (UUID) Namespace*

IETF RFC 4145, *TCP-Based Media Transport in the session Description Protocol (SDP)*

IETF RFC 4234, *Augmented BNF for Syntax Specifications: ABNF*

IETF RFC 4566, *SDP: session Description Protocol*

IETF RFC 4588, *RTP Retransmission Payload Format*

IETF RFC 4605, *Internet Group Management Protocol (IGMP) / Multicast Listener discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")*

IETF RFC 4627, *The application/json Media Type for JavaScript Object Notation (JSON)*

IETF RFC 4629, *RTP Payload Format for ITU-T Rec. H.263 Video*

IETF RFC 4749, *RTP Payload Format for the G.729.1 Audio Codec*

IETF RFC 4756, *Forward Error Correction Grouping Semantics in session Description Protocol*

IETF RFC 4787, *Network Address Translation (NAT) Behavioural Requirements for Unicast UDP*

IETF RFC 4825, *The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)*

IETF RFC 4867, *RTP Payload Format and File Storage Format for the AMR and AMR-WB Audio Codecs*

IETF RFC 4961, *Symmetric RTP/RTCP Control Protocol (RTCP)*

IETF RFC 4975, *The message session Relay Protocol (MSRP)*

IETF RFC 5404, *RTP Payload Format for G.719*

IETF RFC 5626, *Managing Client-Initiated Connections in the session Initiation Protocol (SIP)*

IETF RFC 5627, *Obtaining and Using Globally Routable User Agent URIs (GRUU) in the session Initiation Protocol (SIP)*

IETF RFC 5874, *An Extensible Markup Language (XML) Document Format for Indicating A Change in XML Configuration Access Protocol (XCAP) Resources*

IETF RFC 5875, *An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Diff Event Package*

IETF RFC 6080, *A Framework for session Initiation Protocol User Agent profile Delivery*

IETF RFC 6086, *Session Initiation Protocol (SIP) INFO Method and Package Framework*

IETF RFC 6284, *Port Mapping Between Unicast and Multicast RTP sessions*

3GPP TS 23.228, *IP Multimedia Subsystem (IMS); Stage 2*

3GPP TS 24.109, *Bootstrapping interface (Uu) and network application function interface (Uu); Protocol details*

3GPP TS 24.229, *3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 10)*

3GPP TS 24.237, *IP Multimedia (IM) Core Network (CN) subsystem IP Multimedia Subsystem (IMS) service continuity; Stage 3*

3GPP TS 26.114, *IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction*

3GPP TS 26.234, *Transparent end to end packet switched streaming service (PSS) – Protocols and Codecs*

3GPP TS 26.237, *IP Multimedia Subsystem (IMS) based Packet Switch Streaming (PSS) and Multimedia Broadcast/Multicast service (MBMS) User service; Protocols (Release 8)*

3GPP TS 32.299, *Telecommunication management; Charging management; Diameter charging applications (Release 8)*

3GPP TS 33.203, *3G security; Access security for IP-based services*

3GPP TS 33.220, *Generic authentication Architecture (GAA); Generic bootstrapping architecture*

Broadband Forum TR-069, *CPE WAN Management Protocol*

Broadband Forum TR-069 Amendment 2, *CPE WAN Management Protocol v1.1*

Broadband Forum TR-098, *Internet Gateway Device Data Model for TR-069*

Broadband Forum TR-104, *DSLHome™ Provisioning Parameters for VoIP CPE*

Broadband Forum TR-106, *Data Model Template for TR-069 Enabled Devices*

Broadband Forum TR-135, *Data Model for a TR-069 Enabled STB*

Consumer Electronics Association CEA-2014-A, July 2007, *Web-based Protocol Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE), including the August 28, 2008 Errata*

GSMA PRD IR.84, *Video Share Interoperability Specification, 2.0*

OMA OMA-TS-SIMPLE_IM-V1_0-20100322-C, *Instant Messaging using SIMPLE*

OMA OMA-ERP-Presence_SIMPLE-V1_1-20080627-A, *Presence SIMPLE Specification*

UPnP Forum, *UPnP Device Architecture*

U.S. Department of Commerce/National Institute of Standards and Technology, FIPS PUB 180-1, *Secure Hash Standard*

ECMA, ECMA-262, *ECMAScript® 2015 language specification*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62766-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

native HNI-IGI function

procedures for interactions on the HNI-IGI interface, provided as part of the OITF implementation, typically in native code

3.1.2

non-native HNI-IGI function

procedures for interactions on the HNI-IGI interface, provided by a service provider in JavaScript¹ as part of a DAE application

3.2 Abbreviated terms

For the purposes of this document, the abbreviations given in IEC 62766-1, as well as the following apply.

FCC	Fast Channel Change
GSMA	GSM Association
ISC	IMS service Control
PCh	IMS service Control
PPV	Pay Per View
RET	Re-transmission Function
IETF RFC	Request For Comments
XCAP	XML Configuration Access Protocol
XDM	XML Document Management

4 Interface definitions

4.1 Consumer network to provider network interfaces (UNI)

The functional entities, functions and reference points defined for the consumer network are described in Annex B of IEC 62766-1:2017. Table 1 lists the UNI reference points along with the protocols that have been adopted for their implementation.

¹ JavaScript is a programming language that has been standardised by ECMA as ECMAScript® and as ISO/IEC 16262. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

Table 1 – UNI reference points and protocols

Reference point	Description	Protocols
UNIP-1	Reference point for user initiated IPTV service profile management.	HTTP, XCAP
UNIP-2	Reference point for user initiated profile management of Person-to-Person Communication Enablers, such as presence privacy, resource list management, group management, etc. Note that group management is included to support the management of pre-defined groups that can be reused for several purposes, such as presence privacy, presence request, massaging, chatting, etc.	OMA XDM, OMA Presence Enablers, IMS SIP
UNIS-6	Reference point for user interaction with application logic for transfer of user requests and interactive feedback of user responses (provider specific GUI). HTTP, FLUTE and TCP based application specific protocols are used to interface between the DAE and the IPTV application function.	HTTP, FLUTE
UNIS-7	Requests for transport and encoding of content guide metadata. The reference point includes the metadata and the protocols used to deliver the metadata, and shall be based on DVB-IP BCG (ETSI TS 102 539).	HTTP, DVBSTP
UNIS-8	Authentication and session management relying on IMS.	IMS SIP
UNIS-9	Authentication for GBA Single-Sign on. See IEC 62766-7.	HTTP
UNIS-11	Reference point for control of real time streaming (e.g. control for pause, rewind, skip forward). The reference point includes content delivery session setup when not relying on IMS.	RTSP
UNIS-12	Reference point between the AG and the provider specific application functional entity. Encompasses two functions: Signalling and download of applications in a generic format. This function is subject to standardization. Interaction of generic applications with the provider network. This function is not is not intended to be standardized.	HTTP, FLUTE
UNIS-13	User stream control for multicast of real time content and data.	IGMP
UNIS-14	Reference point used for authorization of service access. See IEC 62766-7.	HTTP
UNIS-15	Reference point to the IPTV service discovery functional entity (FE) to obtain information about IPTV services offered by an IPTV service provider.	HTTP, DVBSTP
UNIT-16	Reference point used for network attachment. Functions connected to this reference point include DHCP server and DHCP relay.	DHCP
UNIT-17	Content stream including content; content encryption (for protected services) and content encoding. This reference point may be used for both multicast and unicast (UNIT-17M and UNIT-17U, respectively).	RTP, HTTP, UDP
UNIT-18	Performance monitoring interface for reporting the performance monitoring results, including RTCP for RET/FCC requesting/control.	RTCP, RTSP
UNIT-19	Multicast Data Channel, used to deliver data of different kinds to the OITF by means of multicast. This reference point can carry discrete data that is carried over unicast through e.g., the interfaces UNIS-6 and UNIS-7. Other uses such as UNI-RMS are not excluded.	FLUTE
UNIS-19	Reference point to the IPTV service provider discovery functional entity to obtain the list of service providers, and related information.	HTTP
UNI-RMS	Remote management using Broadband Forum TR-069 framework and related extensions based on the DVB-IP-RMS specification.	HTTP/TR-069
UNIS-CSP-T	Rights management for protected content, including key management and rights expression. See IEC 62766-7.	HTTP/MARLIN
UNIS-CSP-G	Reference point to support a service and content protection solution specific to the IPTV service provider. This interface may be used to obtain licenses for purchases/subscribed content, control content and the service protection system and may also be used to deliver content.	

Table 2 lists the interfaces apart from the UNI along with the protocols that have been adopted for their implementation.

Table 2 – Other interfaces

WAN gateway LAN interfaces	Interface between OITF/IG and AG and the WAN gateway	DHCP, IGMP
HNI-DM	Interface between the WAN gateway and the OITF to support remote management of the OITF in the home.	UPnP DM UPnP device architecture
HNI-IGI	Interface between the IMS gateway and the OITF providing, to the OITF, access to IG functions for the adaptation to IPTV services on managed networks relying on IMS.	HTTP, SIP

In a device that implements both the OITF and the IG, the use of the HNI-IGI interface is optional. In this case, the device shall support the UNIS-8, UNIS-9 and UNI-RMS interfaces.

The HNI-IGI interface consists of a set of interactions between the OITF and the IG.

The HNI-IGI interface supports two OPTIONAL protocols:

- an HTTP option, where HTTP is deployed over the interface;
- a SIP option, where SIP as defined in IETF RFC 3261 and pertinent IETF RFC extensions is deployed over the interface.

The functionality supported by the OITF and the IG is identical for both options. All IMS specifics are supported by the IG in both options as well.

For the HTTP option, certain interactions on the HNI-IGI interface may be implemented either natively or as a DAE application, whereas other interactions cannot be implemented as DAE applications and shall be implemented in native code. An OITF is said to implement the "native HNI-IGI HTTP option function" if it supports at least (but is not limited to) the interactions that shall be implemented in native code. The case where no native interaction is supported is hereafter known as "non-native HNI-IGI HTTP option function".

The interactions that shall be implemented natively consist of user registration (subclauses 6.4.6.1 and 7.2.4.2.2) including service provider discovery (subclause 6.4.1.1), and GBA procedures (subclause 6.4.6.2) performed at OITF startup.

An OITF that supports the non-native HNI-IGI HTTP option function can still be used in a managed network scenario, but without the support of GBA based authentication or HTTP digest authentication using IG to application servers.

Note that GBA authentication can be achieved using either the GBA authentication using IMS Gateway procedure, specified in IEC 62766-7:2017, 5.4.5 or the, more general, procedure, HTTP Digest authentication using IMS Gateway in IEC 62766-7:2017, 5.4.4. The latter, more general procedure, allows the use of different authentication mechanisms in a way that is transparent to the OITF, including possible future authentication mechanisms, and should preferably be used. It is expected that GBA authentication using IMS gateway procedure will be deprecated and removed in future versions of this specification.

4.2 Provider network reference points description

Table 3 lists the functional entities and reference points in the service provider network.

Table 3 – NPI reference points and protocols

Reference point	Description	Protocols
NPI-1	Reference point between the service access authentication FE and the user database.	Not specified
NPI-2	An optional reference point allowing interaction between IPTV applications and the IPTV control FE.	Not specified
NPI-3	The reference point between authentication session management and person-to-person communication enablers.	ISC interface (ETSI TS 124 503)
NPI-4	Reference point for routing of IPTV service related messages to the IPTV control FE.	ISC interface (ETSI TS 124 503)
NPI-6	This reference point allows the IPTV control FE to retrieve the subscriber's IPTV-related service data when a user registers in the IMS network.	Not specified
NPI-7	This reference point allows person-to-person application enablers to retrieve the subscriber's IMS data from the user database.	Sh interface (ETSI TS 129 329)
NPI-9	This reference point allows the IPTV control point to retrieve the subscriber's IMS-specific data from the user database.	Sh interface (ETSI TS 129 329)
NPI-10	A reference point for the allocation/de-allocation and control of content for a unicast session.	RTSP
NPI-11	A reference point for sending events and charging information.	Rf and Ro (ETSI ES 282 010)
NPI-12	This reference point allows the authentication and session management FE to retrieve the subscriber's IMS data from the user database as a part of the user's IMS registration.	Cx (ETSI TS 183 033)
NPI-14	A reference point from charging FE and authentication and session management FE.	Rf (ETSI ES 282 010)
NPI-15	This reference point controls the resources and admission control.	Gq' (ETSI TS 183 017)
NPI-16	Reference point between the transport processing function and resource and admission control.	Re (ETSI TS 183 060)
NPI-17	Reference point between the IPTV applications and the IPTV service profile.	XCAP
NPI-18	Reference point between the service access and authentication FE and the IPTV applications.	Not specified
NPI-19	This reference point shall be used for unicast session setup control between the authentication and session management and the content delivery network controller.	SIP/SDP
NPI-20	This optional reference point allows the retrieval of CG data.	Not specified
NPI-21	This reference point allows the GBA single sign-on functional entity to validate user credentials.	Not specified
NPI-25	This reference point allows forwarding unicast control messages to the appropriate content delivery network controller FE.	SIP/SDP
NPI-26	The reference point allows the content delivery network controller to delegate the handling of a unicast session to a specific cluster controller.	SIP/SDP
NPI-27	The reference point between the authentication proxy and the GBA single sign-on node allows the proxy to retrieve a user key for authentication purposes.	Not specified
NPI-28	This reference point shall be used to push the user access capabilities to the network attachment and the RAC.	ETSI ES 283 034
NPI-30	This reference point supports the IPTV service provider discovery step of the service discovery procedure relying on IMS.	ISC interface (ETSI TS 124 503)
NPI-32	Reference point between the ASM FE and the IMS messaging AS. (This is the ISC interface defined in 3GPP TS 23.228)	ISC interface (ETSI TS 124 503)
NPI-33	Reference point allowing interaction between IPTV applications and the IPTV metadata control FE. This is not subject to standardization.	Not specified

Reference point	Description	Protocols
NPI-34	The reference point between the IMS messaging server and the notification services. It is based on IMS SIP as defined in 3GPP TS 24.229.	SIP
NPI-36	This reference point allows access to notification services. It is based on Parlay X -API as defined by (http://www.3gpp.org/ftp/Specs/html-info/29-series.htm).	Parlay X
NPI-38	This reference point between notification services and multicast and delivery control function supports multicast traffic for emergency services and is FFS.	Not specified
NPI-39	This reference point between emergency services and the notification services is local and regional specific.	Not specified
NPI-40	Content delivery function (CDF) stream control for multicast of real time. The protocol used on this interface is IGMP. This interface is optional.	IGMP
NPI-41	Content stream including content; content encryption (for protected services) and content encoding. This reference point is used for multicast delivery. The protocol used on this interface is RTP. This interface is optional.	RTP
NPI-42	This reference point between the IPTV application and the multicast content delivery function supports multicast traffic for notification services.	FLUTE
NPI-CSPT1	Reference point to confirm whether a Marlin content license can be issued for the request received via UNIS CSP-T.	See IEC 62766-7
NPI-CSPG1	Reference point to allow the CSP-G server to be provisioned with entitlement information by IPTV applications.	Not specified
NPI-CSPG1a	Reference point to allow the CSP-G server to be provisioned with entitlement information by the IPTV service profile.	Not specified
NPI-CSPG3	Reference point for the key management function to exchange content encryption information with CSP-G server.	HTTP
NPI-CSPT1a	Reference point used by the Marlin DRM system to include business information or a reference to business information into a DRM request (e.g. license request) as requested via UNIS-CSP-T, and the subsequent confirmation and retrieval of this business information when the DRM request is consumed.	See IEC 62766-7
NPI-CSPT2	Reference point, used in the managed network model, to retrieve information on the appropriate cluster controller in the content delivery network that will serve a particular request for purchased or subscribed-to content. This chosen cluster controller will be contacted by the CSP-T server functional entity via NPI-CSP3.	Not specified
NPI-CSPT3	Reference point to retrieve the appropriate encryption key needed to prepare a Marlin content license for the chosen content. It is the content encryption key for downloadable content or the key that encodes the Marlin short term key message that contains the key that encodes the streaming media.	HTTP
NPI-43	Reference point that provides GBA authentication mechanism to the service access authentication function.	
NPI-44	Reference point where the encrypted content is stored on the content storage entity for delivery by the content delivery function. This interface has been identified just to illustrate informatively the separation between content encryption, which is part of content preparation, and content delivery.	Not specified
NPI-45	Reference point where the content service, programme and content keys and ECM attached information are provided to the CoD encryption management function.	HTTP
NPI-46	Reference point where the content service, programme and content keys and content protection related ECM attached information (e.g. ECM, DRM metadata) are provided to the scheduled content encryption function. This interface is specified by this version of the specification for the unicast stream encryption case. This interface is not specified by this version of the specification for the multicast stream encryption case.	HTTP

Reference point	Description	Protocols
NPI-47	Reference point where the on-demand content is fetched by the content delivery function for delivery. This interface has been identified just to illustrate informatively the separation between content encryption, which is part of content preparation, and content delivery.	Not specified
NPI-48	Reference point for the key management function to provide appropriate information to the IPTV applications functional entity, e.g. in relation with content access licenses.	Not specified
NPI-49	Reference point for the content management function to provide the CoD encryption function with content related information.	Not specified
NPI-50	Reference point for the scheduling function to provide the recording function with a record list/schedule for the catch-up and start-over use cases.	Not specified
NPI-51	Reference point for the content management function to provide appropriate information to the IPTV applications functional entity.	Not specified
NPI-52	Reference point for the scheduling function to provide appropriate information to the IPTV applications functional entity.	Not specified
NPI-53	Reference point for the scheduling function to provide the scheduled content encryption function with content related information and schedule.	Not specified
NPI-AR-01	Reference point for providing static audience data about users who have opted-in. It includes content metadata and user related information stored in the IPTV service profile.	
NPI-AR-02 NPI-AR-02' NPI-AR-02''	Reference points for collecting the information intercepted by the transport processing function, the IPTV control, the cluster control or other FEs based on different criteria, e.g. the events triggered by the audience research collector, event detected from other FEs, the deployment done by the service provider etc. Note that the IPTV control can retrieve the audience research data from the ITF or the cluster controller using existing SIP messages such as SIP INFO, MESSAGE, INVITE or PRESENCE.	
NPI-AR-03	Reference point used for exposing the audience research data to the audience research agency.	

4.3 Interfaces to external systems, consumer network

Table 4 documents the interfaces that are external to the consumer network.

Table 4 – External interfaces from the consumer network

DLNA function	Interface between the OITF and DLNA devices in the home.	DLNA
---------------	--	------

5 Structure of this document

Each subclause of this document identified below defines the procedures that use a specific protocol:

- Clause 6: HTTP
- Clause 7: SIP and SIP/SDP
- Clause 8: RTSP
- Clause 9: IGMP and multicast protocol
- Clause 10: RTP/RTCP
- Clause 11: UPnP
- Clause 12: DLNA

Clause 13: DHCP

Clause 14: UDP

Clause 15: FLUTE

Clause 16: Diameter

The annexes cover the following topics:

Annex A: Example messages (informative)
Annex B: User profile description (informative)
Annex C: Mapping attributes for scheduled content
Annex D: <protocol> names
Annex E: System infrastructure
Annex F: System infrastructure mechanisms (informative)
Annex G: NAT-T informational flows for IPTV services (informative)
Annex H: Port mapping and NAT traversal for FCC/RET for multicast content services
Annex I: Presence XML schema
Annex J: Protocol procedure subclause structure
Annex K: OITF-specific TR-135 and TR-106 remote management objects
Annex L: New event package for SIP SUBSCRIBE/NOTIFY (informative)
Annex M: Overview of notification services in OIPF R2 (informative)
Annex N: Fast Channel Change and Retransmission (FCC/RET)
Annex O: IG handling of IMPUs in association with GRUU (informative)
Annex P: FDT schema extensions
Annex Q: IG Service awareness
Annex R: Definition of content reporting info package
Annex S: Definition of Digital-Media-Purchase info package
Annex T: Definition of parental control watched content info package
Annex U: Common types
Annex V: Schema extension for FLUTE FDT

6 HTTP

6.1 HTTP reference points

This subclause defines the protocol for the use of HTTP over the following reference points:

- HNI-IGI – HTTP option

Certain interactions on the HNI-IGI interface can only be implemented natively, while the rest can be implemented either in native code or in a DAE application. In the following subclauses, if no qualification is provided the function can be performed natively or as a DAE application.

- UNIP-1
- UNIS-6
- UNIS-7
- UNIS-9
- UNIS-15
- UNIT-17

- UNIS-19

6.2 IG as a protocol converter supporting the HNI-IGI interface – HTTP option

In support of the HNI-IGI HTTP option, the IG shall act as a protocol converter between HTTP and SIP. In that respect, the IG acts as an HTTP server towards the OITF, and as a SIP user agent (UA) towards the IMS network. The following lists the behaviour of the IG as a SIP UA towards the IMS network:

- The IG shall handle all SIP headers that are mandatory in IETF RFC 3261; creating and storing them where applicable when initiating requests from the OITF towards the network or when receiving incoming requests from the IMS network targeted for an OITF. The same applies to SIP responses. The different services specify what needs to be stripped before a request/response is sent to the OITF.
- The IG shall handle all IMS specific SIP headers; storing what is received from the network for re-insertion during outgoing requests (e.g. P-Preferred-Identity) according to ETSI TS 124 503, and stripping those headers before a request/response is sent to the OITF.
- The IG shall behave transparently to all SIP headers defined in IETF RFC 3261, that are received from the network, and shall not alter them before being sent to the OITF (encoded as HTTP headers), where applicable. The same applies to SIP headers (encoded as HTTP headers) sent from the OITF to the IMS network.
- The IG shall validate incoming SIP headers (encoded as HTTP headers) from the OITF before accepting a request. The same applies to SIP responses received from the OITF. The various services specify the validation to be performed by the IG.
- The IG shall not validate compliance to any XML schema.
- The IG shall validate SDP (encoded in the HTTP body) syntax for correctness.
- In regard to IMS registration, the IG shall perform the IMS registration when requested by the OITF. The IG is therefore stateful to IMS registration. However, graceful de-registration and re-registration shall be triggered by the OITF. The IG shall deal with all non-graceful circumstances. The specific subclauses on IMS registration specifies detailed behaviour for the OITF and IG.
- The IG shall be stateful, to all IMS sessions. Session initiation, termination, and session refresh can be triggered by the OITF or the network, depending on the specific service. The IG shall deal with non-graceful circumstances.

6.3 Protocols for IPTV service functions

6.3.1 Multicast content streaming with SIP session management protocol over HNI-IGI – HTTP option

6.3.1.1 General

When the OITF initiates, modifies or terminates a multicast content service, the OITF sends HNI-IGI messages containing the appropriate method, mapped to HNI-IGI as described in 6.6.1.

The SIP-specific information in the related messages is described in 7.1.2.1. The SIP-specific information is mapped to the HNI-IGI protocol, as described in 6.6.1. In particular, the OITF creates HTTP headers for an HNI-IGI message by adding “X-OITF-” in front of the necessary SIP header names. In addition, OPTIONAL parameters may be included as defined in ETSI TS 124 503.

Certain interactions on the HNI-IGI interface shall be implemented natively, while the remaining applicable interactions may be implemented either natively or as a DAE application. In 6.3.1.2 to 6.3.1.8, if no qualification is provided, the interaction can be performed natively or as a DAE application.

6.3.1.2 Session initiation

The HNI-IGI function in the OITF shall follow the following procedure for session initiation:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 5

HTTP request body: SDP offer containing the following elements (conforming to ETSI TS 183 063):

- The m-line(s) shall be set to the multicast content service which the OITF intends to join first, according to the mapping defined in Clause C.1.
 - If the multicast content service includes FCC and/or RET and the OITF supports FCC and/or RET, the <proto> field shall be “RTP/AVPF”. If the OITF does not support FCC and/or RET, “RTP/AVPF” shall not be used.
- The c-line(s) shall be set to the multicast content service which the OITF intends to join first, according to the mapping defined in Clause C.1,
- An a=bc_service: BCserviceld line shall indicate the multicast content service that the OITF intends to join (according to the mapping defined in Clause C.1.
- If the OITF does not intend to use FCC and/or RET and has knowledge of the bandwidth of the multicast content service with the highest bandwidth requirement included in the session, the b-line shall be included and set to this value.
 - If the OITF supports FEC and the multicast content service has FEC enabled, then the OITF shall include the additional bandwidth in the value set in the b-line. If the OITF does not support FEC and the multicast content service includes FEC that uses the same multicast group address then the FEC bandwidth shall be included.
 - If the OITF supports Network Generated Notification service and the multicast content service has associated Network Generated Notification service, then the OITF shall include the additional bandwidth in the value set in the b-line. If the OITF does not support Network Generated Notification service and the multicast content service has associated Network Generated Notification service that uses the same multicast group address, then the additional bandwidth shall be included.
- If the OITF intends to use FCC and/or RET, it shall have performed the procedure in 6.3.1.3 and shall include in the session b-line the value returned from that procedure.
- If the OITF intends to make use of RET services only for a RET-enabled multicast content service session, it shall include the attribute a=rtcp-fb:<fmt>nack where <fmt> indicates the RTP payload type of the IP multicast stream that carries the multicast content service.
- If the OITF intends to make use of FCC service for a FCC-enabled multicast content service session, it shall include the attribute a=rtcp-fb:<fmt>nack rai where <fmt> indicates the RTP payload type of the IP multicast stream that carries the multicast content service.
- If the OITF wants to make use of RET and/or FCC, there shall be an a=rtcp:<port> <network type> <address type> <connection address> line after the m-line for the IP multicast stream, providing the OITF destination IP address and port for the RTCP feedback messages associated with the IP multicast stream.

NOTE The RTCP feedback target transport address signalled in the a=rtcp: <port> <network type> <address type> <connection address> line on the media level for the IP multicast stream is only valid for the first IP multicast channel the OITF intends to connect to. For other channels that are part of the same broadcast package, the SDNS signalled values

(RTCPReporting@DestinationPort and RTCPReporting@DestinationAddress) for each particular channel shall prevail.

- An a=recvonly line

In order for the OITF to connect to the FEC stream associated with the original multicast stream, additional parameters shall be included in the SDP offer as follows:

- An m-line for the FEC stream, as indicated by the service discovery or Metadata control FE. The m-line shall be set according to the mapping defined in Clause C.1.
- A c-line according to the mapping defined in Clause C.1.

In order for the OITF to consume additional network generated notification service associated with the multicast content service, additional parameters shall be included in the SDP offer as follows:

- An m-line for the notification multicast stream, as indicated by the service discovery or Metadata control FE. The m-line shall be set according to the mapping defined in Clause C.1.
- A c-line according to the mapping defined in Clause C.1.
- An a=bc_service: BCserviceId line shall indicate the Network Generated Notification service that the OITF intends to join (according to the mapping defined in Clause C.1).
- Optionally one or more a=bc_service_package: <BCPackageId> as defined in Clause C.2. The initial offer shall not contain mult_list and bc_tv_service_id_list parameter. If the initiation is the result of a previously denied initiation, the OITF may restrict the Network Generated Notification services by including mult_list attributes.

If the OITF indicates its support for RET by means of the "a=rtcp-fb:<fmt> nack" and multicast RET is offered as a service as indicated by SD&S, in order for the OITF to connect to the multicast RET stream, the SDP offer shall include an additional m-line for the multicast RET stream (m=<media> <port> <proto> <fmt>) that shall be set according to the mapping defined in Clause C.1.

Step 2: If the request is for a multicast content service session, the IG shall validate that the request includes all the mandatory SIP headers for the process as per Table 5. The IG shall send a SIP INVITE to the network to request the initiation of the multicast content service session, and shall wait for the response to the request. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: On receipt of the response from the network the IG shall return a HTTP 200 OK response (or other appropriate received responses) to the OITF to report the response to the initiation request. The response shall include a list of SIP headers as per Table 6 in addition to the normal HTTP headers as per IETF RFC 2616, and the same SDP answer body that was received by the IG in the SIP message.

Step 4: When the OITF receives the response to the INVITE, it shall examine the media parameters in the received SDP. The OITF shall restrict the multicast content services that it joins according to the parameter (a=bc_service_package attribute) received from the IPTV control FE. However, if the OITF retrieved the IPTV user profile prior to session initiation, then it may ignore the a=bc_service_package attribute.

If the OITF receives an error code with an Insufficient Bandwidth indication in the response from the IG, the OITF may perform a new INVITE with a reduced maximum bandwidth for the multicast content service. This procedure may be repeated. If no agreement can be reached, the OITF may display a failure message to the user.

Step 5: Upon receipt of a 200 OK response, the OITF shall send an HTTP PENDING_IG to acknowledge the final response. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616 RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 7

HTTP request body: Empty

Table 5 – Supported HTTP extension headers in the HNI-IGI INVITE request message for multicast content service session setup (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INVITE request shall be the well known PSI (Public service Identifier) of the content service: OIPF_IPTV_SC_service@<domain name>. The domain part shall be the IPTV service provider domain name obtained via service provider discovery.	IETF RFC 3261 INVITE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Contact The URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261 (application/sdp)
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Supported	IETF RFC 3261 set to timer
X-OITF-session-Expires	IETF RFC 4028
X-OITF-Recv-Info shall be empty or the list of info packages the OITF is willing to receive	IETF RFC 6086 ETSI TS 26.237

Table 6 – Supported HTTP extension headers in the response message to an HNI-IGI INVITE request message for multicast content service session setup (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-session-Expires	IETF RFC 4028
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Recv-Info shall be empty to stop receiving any SIP INFO including any Info Package Or shall be set to Content-Reporting Info Package to indicate willingness to receive the Info Package. And/Or shall be set to CoD-Bookmark Info Package according to subclause 12 of ETSI TS 26.237 to indicate willingness to receive the Info Package And/Or shall be set to Digital-purchase Info Package to indicate willingness to receive the Info Package	IETF RFC 6086 ETSI TS 26.237

Table 7 – Supported HTTP extension headers in the HNI-IGI ACK message for a successful multicast content service session setup (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message	IETF RFC 3261 ACK <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line" of the initial request	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what has been inserted in the INVITE message. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261

6.3.1.3 Retrieval of bandwidth parameter for FCC and/or RET enabled multicast content service

If the OITF intends to use FCC and/or RET, when the multicast content service is FCC and/or RET enabled, the OITF shall use the procedure defined in 6.3.2.1.1, with the following modifications.

- The request-URI in the method line is set to well-known PSI for the multicast content service.
- The OITF shall include an HTTP body that shall include an SDP that includes one m line that matches the m line associated with the multicast content service the OITF intends to join first (see 6.3.1.2) with the following exceptions:
 - no b-line is included;
 - The following additional a attribute shall be included when the OITF intends to use RET:
 - a=rtcp-fb:<fmt> nack
 where <fmt> indicates the RTP payload type of the IP multicast stream that carries the content service.
- The following additional a attribute shall be included when the OITF intends to use FCC:
 - a=rtcp-fb:<fmt> nack rai
 where <fmt> indicates the RTP payload type of the IP multicast stream that carries the content service.

The returned response shall include total bandwidth of the multicast content service with the highest total bandwidth, including the overhead bandwidth required for the FCC/RET stream. Maximum bandwidth will be signalled by means of the following media level bandwidth modifier:

- b=AS:<bandwidth>

where <bandwidth> is the maximum calculated bandwidth according to the b=AS bandwidth modifier defined in IETF RFC 4566, expressed in kbit/s.

6.3.1.4 Session modification

To join a service outside the set of channels negotiated at session initiation or to perform a bandwidth modification, the OITF shall send a request to the IG for session modification. The OITF shall generate a re-INVITE request, as defined in Table 5.

The OITF shall include an SDP offer in the session modification request. The format of this request shall be the same as for a session initiation.

6.3.1.5 Session termination

To terminate a multicast content streaming session, the OITF shall use the following procedure:

- Step 1:** The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:
- HTTP request header: Including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 8
- HTTP request body: Empty
- Step 2:** The IG shall validate that the request includes all the mandatory SIP headers needed for the message as per Table 8. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection. The IG shall send a SIP BYE to the network, to request

the termination of the multicast content streaming session, and shall wait for the response.

Step 3: The IG shall then return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the Termination request. The response shall include, in addition to the normal HTTP headers as per IETF RFC 2616, a list of SIP headers as per Table 9.

Table 8 – Supported HTTP extension headers in HNI-IGI BYE request for teardown of a multicast content streaming session (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI shall be set to the contact returned in the 200 OK for the invite.	IETF RFC 3261 BYE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line" of the initial request	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

Table 9 – Supported HTTP extension headers in the response to an HNI-IGI BYE request for teardown of a multicast content streaming session (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.1.6 Session refresh

It is the responsibility of the OITF application to refresh the multicast content streaming session before the session expires. The IG shall consider a session terminated if it is not refreshed.

6.3.1.7 Content reporting and management of content reporting of watched scheduled content

6.3.1.7.1 Content reporting by the OITF

The OITF shall follow the following procedure for reporting a watched scheduled content:

Step 1: To report the scheduled content being watched after the IPTV end user stopped zapping for the configured time, and if the IPTV control FE permitted content reporting, the OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616

- <list of SIP headers encoded as HTTP headers> – see Table 10

HTTP request body: shall include the XML document as per Annex D of ETSI TS 126 237

The Content-Reporting Info Package shall contain an XML document defined in Annex D of ETSI TS 126 237 with MIME type “application/3gpp-ims-pss-mbms-command+xml”.

The XML document shall include either one element of a PSS content switch data or one element of MBMS content switch data. The selection of PSS content switch data or MBMS content switch data is determined according to the content delivery mechanism. Scheduled Content delivered to the OITF through a multicast mechanism shall use the MBMS content switch choice. Scheduled Content delivered by unicast streaming mechanism shall use the PSS content switch choice. For either of those elements there shall be at least one defined sequence included.

The elements in the XML document shall be populated as follows:

- For PssSwitchData
 - ContentID – RTSP URI of the content
 - DateTime – may be used
- For MbmsSwitchData
 - serviceld – The BCserviceld of the channel
 - Programmeld – Shall not be used
 - DateTime – may be used

Also note that the SIP INFO including the info-event Watched Content is sent only within the context of a scheduled content session.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the process as per Table 10. The IG shall send a SIP INFO to the network and shall wait for the response to the request. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: On receipt of the response from the network, the IG shall return a HTTP 200 OK response (or other appropriate received responses) to the OITF to report the response to the INFO request. The response shall include a list of SIP headers as per Table 11 in addition to the normal HTTP headers as per IETF RFC 2616.

6.3.1.7.2 Management of reporting by IPTV control FE

The OITF shall comply with the following when it comes to management of reporting a watched scheduled content:

Step 1: At any time, the IG can receive a SIP UPDATE, including an empty Recv-Info header or including the Content-Reporting Info Package, from the network to order the OITF to stop or start reporting the watched content. When a SIP UPDATE is received by the IG, the IG shall return an HTTP 200 OK response to the OITF. The response includes a list of SIP headers as per Table 12 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 2: Once the OITF accepts the incoming SIP UPDATE, it shall send to the IG an HTTP HNI-IGI PENDING_IG request to acknowledge the receipt of the SIP UPDATE. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 13

HTTP request body: Empty

If the SIP UPDATE is for resumption of content reporting, the OITF shall immediately report the currently watched content in accordance with 6.3.1.7.1.

Table 10 – Supported HTTP extension headers in the HNI-IGI SIP INFO request message for content reporting (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INFO request shall be the well known PSI (Public service Identifier) of the content service: OIPF_IPTV_SC_service@<domain name>.	IETF RFC 3261 INFO <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what was used in the SIP INVITE for the scheduled content session. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
X-OITF-Call-ID shall match what was used in the SIP INVITE for the scheduled content session	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF_Info-Package shall be set to Content-Reporting Info Package	IETF RFC 6086
X-OITF-Content-Type shall be set to "application/3gpp-ims-mbms-psscommand+xml" that corresponds to Annex D of ETSI TS 26.237	IETF RFC 6086 3GPP TS 26.237
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Content-Disposition	IETF RFC 6086

Table 11 – Supported HTTP extension headers in the response message to an HNI-IGI INFO request message for content reporting (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

**Table 12 – List of HTTP extension headers
for an incoming SIP UPDATE (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INVITE shall match the contact URI included in the contact filed of the SIP INVITE for the scheduled content session	IETF RFC 3261 UPDATE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact The URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 6086
X-OITF-Content-Length	IETF RFC 3261
Recv-Info shall be set to remove support for the reception of the Content-Reporting Info Package OR shall be set to indicate support for the reception of Content-Reporting Info Package	IETF RFC 6086

**Table 13 – Supported HTTP extension headers in the response message
to an incoming HNI-IGI SIP UPDATE request message (OITF→IG)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.1.8 Network-based multicast content streaming time shift**6.3.1.8.1 User-initiated activation of multicast content streaming time shift**

As a prerequisite, it is assumed that the user has an established multicast content streaming session as per the procedure defined in 6.3.1.2. The OITF shall follow the following procedure to initiate a time shift for the watched multicast content service:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 5.

HTTP request body: the request body shall include a MIME Multipart/Related container conforming to IETF RFC 2387. The first body part shall be an SDP offer containing the following elements conforming to ETSI TS 183 063:

- An m-line for the RTSP control channel set in accordance with 6.3.2.1.2.
- An m-line for the unicast media delivery channel set in accordance with 6.3.2.1.2, and where the transport and codec values shall be set identical to the values used in the multicast content streaming setup request, while following the rules for modifying a session in IETF RFC 3264. The multicast media descriptor shall be de-activated. Note that the media descriptors shall be reused to avoid the SDP from eventually becoming too large.
- BCserviceld shall be present only if the OITF has not informed the IPTV control FE of the selected channel prior to this procedure (as defined in 6.3.1.7) and shall be set to the value of the current channel.

The second body part shall conform to the OITF-IPTV service Action command XML schema in 6.3.1.8.3, and shall only include the following:

- IPTVActionDataCommand shall be set to "SwitchtoTM".
- SwitchToTM shall be set to elements defined by "IPTVBcActionData".

Step 2: The IG shall validate that the request includes all the mandatory SIP headers as per Table 5. The IG shall send a SIP re-INVITE to the network to request the initiation of the time shift procedure, and shall wait for the response to the request. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: On receipt of the response from the network the IG shall return a HTTP 200 OK response (or other appropriate received responses) to the OITF to report the response to the time shift activation request. The response shall include a list of SIP headers as per Table 6, in addition to the normal HTTP headers as per IETF RFC 2616. The SDP answer shall be in accordance with 7.2.3.2.2.5 in addition to the following:

- An a=fmtp:iptv_rtsp h-offset shall be included in the SDP answer, and shall have a value different than 0 indicating the offset in the time shifted selected content. This can be used by the OITF during media control.

Step 4: When the OITF receives the response to the re-INVITE, it shall examine the media parameters in the received SDP, and shall store the parameters a=fmtp:iptv_rtsp h-session, a=fmtp:iptv_rtsp h-offset, and a=fmtp:iptv_rtsp h-uri for later usage.

Step 5: Upon receipt of a 200 OK response, the OITF shall send an HTTP HNI-IGI PENDING_IG request to acknowledge the final response. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 7

HTTP request body: Empty

6.3.1.8.2 User initiated de-activation of multicast content steaming time shift

As a prerequisite it is assumed that the user has an activated scheduled content time shift session as per the procedure in 6.3.1.8.1.

The OITF shall follow the following procedure to initiate the de-activation of a time-shifted multicast content service:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 5.

HTTP request body: the request body shall include a MIME Multipart/Related container conforming to IETF RFC 2387. The first body part shall be an SDP offer identical to the one that activated the scheduled content time shift with the following exceptions:

- The m-line for the multicast content streaming session shall be reactivated (i.e. the port shall not be set to 0) while following the rules for modifying a session in IETF RFC 3264. The unicast media delivery channel and the RTSP control channel shall be de-activated. Note that the media descriptors shall be reused to avoid the SDP from eventually becoming too large.
- BCserviceld shall be set to the value of the selected channel.

The second body part shall conform to the OIPF-IPTV service Action command schema in 6.3.1.8.3, and shall only include the following:

- IPTVActionDataCommand shall be set to "SwitchtoBC".
- SwitchToBC shall be set to elements defined by "IPTVBcActionData".

Step 2: The IG shall validate that the request includes all the mandatory SIP headers as per Table 5. The IG shall send a SIP re-INVITE to the network to request the de-activation of the time shift procedure, and shall wait for the response to the request. The IG shall reject a request that is missing any mandatory SIP headers, with a non-200 OK HTTP response, including the reason for rejection.

Step 3: On receipt of the response from the network, the IG shall return a HTTP 200 OK response (or other appropriate received responses) to the OITF to report the response to the time shift de-activation request. The response shall include a list of SIP headers as per Table 6, in addition to the normal HTTP headers as per IETF RFC 2616.

Step 4: Upon receipt of a 200 OK response, the OITF shall send an HTTP HNI-IGI PENDING_IG request to acknowledge the final response. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 7

HTTP request body: Empty

6.3.1.8.3 XML schema for OITF-IPTV commands

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:IPTVAction:2009"
  xmlns:tns="urn:oipf:iptv:IPTVAction:2009"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="urn:org:etsi:ngn:params:xml:ns:iptvbcserviceactiondata"
  xmlns:co="urn:org:etsi:ngn:params:xml:ns:iptvcodserviceactiondata"
  xmlns:np="urn:org:etsi:ngn:params:xml:ns:iptvnpvrserviceactiondata"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:import
    namespace="urn:org:etsi:ngn:params:xml:ns:iptvbcserviceactiondata"
    schemaLocation="imports/iptvbcserviceactiondata.xsd"/>

  <xs:import
    namespace="urn:org:etsi:ngn:params:xml:ns:iptvcodserviceactiondata"
    schemaLocation="imports/iptvcodserviceactiondata.xsd"/>

  <xs:import
    namespace="urn:org:etsi:ngn:params:xml:ns:iptvnpvrserviceactiondata"
    schemaLocation="imports/iptvnpvrserviceactiondata.xsd"/>
```

```

<xs:element name="IPTVAction" type="tns:IPTVActionType"/>

<xs:complexType name="IPTVActionType">
  <xs:choice>
    <xs:element name="Notify" type="tns:NotifyType"/>
    <xs:element name="Record" type="tns:RecordType"/>
    <xs:element name="SwitchToTM" type="tns:SwitchToTMType"/>
    <xs:element name="SwitchToBC" type="tns:SwitchToBCType"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="NotifyType">
  <xs:choice>
    <xs:element ref="bc:IPTVBcActionData" />
    <xs:element ref="co:IPTVCoDActionData" />
    <xs:element ref="np:IPTVNpvrActionData" />
  </xs:choice>
</xs:complexType>
<xs:complexType name="RecordType">
  <xs:choice>
    <xs:element ref="bc:IPTVBcActionData" />
    <xs:element ref="np:IPTVNpvrActionData" />
  </xs:choice>
</xs:complexType>
<xs:complexType name="SwitchToTMType">
  <xs:choice>
    <xs:element ref="bc:IPTVBcActionData" />
  </xs:choice>
</xs:complexType>
<xs:complexType name="SwitchToBCType">
  <xs:choice>
    <xs:element ref="bc:IPTVBcActionData" />
  </xs:choice>
</xs:complexType>
</xs:schema>

```

6.3.2 Unicast content streaming with SIP session management

6.3.2.1 Protocol for SIP session management over HNI-IGI – HTTP option

6.3.2.1.1 Retrieval of session parameters

If the OITF does not have all the necessary parameters to form the SDP offer, the HNI-IGI function in the OITF shall retrieve missing SDP parameters using the following procedure:

Step 1: The OITF shall send an HTTP POST request to the IG on the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 14

HTTP request body: empty.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers required for the outgoing message as per Table 14. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response that includes the reason for rejection.

Step 3: The IG shall send a SIP OPTIONS message to the network, to retrieve missing SDP parameters and shall wait for the response to the request. The IG shall then return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the request for missing SDP parameters. The response

includes a list of SIP headers as per Table 15, in addition to the normal HTTP headers as per IETF RFC 2616, as well as an SDP body containing the missing SDP parameters according to 7.2.3.2.1.2.

Table 14 – Supported HTTP extension headers in HNI-IGI OPTION request for unicast content streaming session setup parameters (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request URI shall be set to the PSI (Public service Identifier) of the content service as follows: For CoD: OIPF_IPTV_CoD_service_ *@<domain name> Where: The wild card part (*) is a content instance identifier, constructed according to 4.3.2.2 in IEC 62766-3 when CoD content identifiers are delivered via the Content Guide. For DAE applications signalling CoD, the wild card part is constructed according to 8.2.2 in IEC 62766-5-1. The domain part (<domain name>) is the IPTV service provider domain name, obtained from the IPTV service provider discovery function.	IETF RFC 3261 OPTIONS <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line OPTIONS" header	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Accept	Set to application/sdp as per IETF RFC 3261
X-OITF-Recv-Info shall be set to remove support for the reception of the CoD-Bookmark Info Package Or shall be set to indicate support for the reception of the CoD-Bookmark Info Package NOTE The CoD-Bookmark Info Package is defined in subclause 12 of 3GPP TS 26.237	IETF RFC 6086

Table 15 – Supported HTTP extension headers in the response to an HNI-IGI OPTION request for unicast content streaming session setup parameters

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

6.3.2.1.2 Session initiation

The OITF shall initiate the request for a unicast content streaming session using the following procedure.

Step 1: The OITF shall send an HTTP POST request to the IG on the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 16

HTTP request body: The request body includes the SDP offer generated by the OITF. The SDP offer shall include a media description for the RTSP content control channel and the media description for the content delivery channel. SDP shall be used as specified in ETSI TS 124 503.

- SDP Parameters for the RTSP control channel

The RTSP content control media description shall be carried by TCP and follow IETF RFC 4145. Hence, the SDP parameters for the RTSP content control channel shall be set as follows:

- An m-line for an RTSP stream of format: m=<media> <port> <transport> <fmt>
 - The <media> field shall have a value of "application".
 - The <port> field shall be set according to IETF RFC 4145. The "a=setup" attribute shall be set to 'active', and port field shall be set to a value of "9", which is the discard port.
 - The <transport> field shall be set to "TCP" or "TCP/TLS". The former shall be used when RTSP runs directly on top of TCP and the latter shall be used when RTSP runs on top of TLS, which in turn runs on top of TCP.
 - The <fmt> parameter shall be included and set to "iptv_rtsp" (ex. m=application 9 tcp iptv_rtsp)
- An "a=setup" attribute shall be present and set to "active" as defined in IETF RFC 4145 (e.g. a=setup:active).
- An "a= connection" attribute shall be present and set as "new" as defined in IETF RFC 4145 (e.g. a=connection:new).
- A c-line shall include the network type with the value set to "IN", the address type set to "IP4" and IP address of the RTSP content control stream (e.g. c=IN IP4 <IP_ADDRESS>).

One or more a=fmtp lines representing RTSP specific attributes set as follows:

- The RTSP version may be specified in a "fmtp:iptv_rtsp version" parameter.

- SDP Parameters for the content delivery channels

For each media stream controlled by the RTSP content control channel, the SDP offer shall include a content delivery channel media description, set as follows:

- The m-line indicates the type of the media ("video"), the transport protocol and the OITF's desired port of the related content delivery channel as follows: m=<media> <port> <proto> <fmt>
- <fmt> settings:
 - When MPEG2-Transport Stream ISO/IEC 13818-1:2015 is used, <fmt> shall be "33" as specified in IETF RFC 3551
 - When OPTIONAL Timestamped-TS defined by IEC 62481-2 is used, the RTP/AVP dynamic payload type shall be used and <encoding name> of "a=rtpmap" line shall be "vnd.dlna.mpeg-tts" as specified in IEC 62481-2,

e.g.

m=video 49232 RTP/AVP 98

a=rtpmap:98 vnd.dlna.mpeg-tts/27000000

- When the media stream is retransmission protected, <fmt> shall have two payload type values: one for the original stream (<fmt1>) and one for the retransmission stream (<fmt2>), with the two streams being SSRC multiplexed. E.g.
m=video <port> RTP/AVPF <fmt1><fmt2>

The <fmt> values are retrieved by the OITF from the response to the SIP OPTIONS and the following attributes are used that define which <fmt> corresponds with which stream shall be used:

a=rtcp-fb:<fmt1> nack

a=rtpmap:<fmt2> <rtx/clock_rate>

a=fmtp:<fmt2> apt= <fmt1>

Where <fmt2> is the RTP payload type of the retransmission stream, and <fmt1> is the RTP payload type of the original media stream.

- <proto> settings:
 - <proto> shall be set according to information obtained by the OITF either by OPTIONS or in the service access stage. If streaming is RTP, <proto> shall be set to RTP/AVP. If streaming is RTP and retransmission is supported, <proto> shall be set to RTP/AVPF. If streaming is direct over UDP, <proto> shall be set to “MP2T/H2221/UDP” or “RAW/RAW/UDP”
- The “c- line” shall include the network type with the value set to “IN”, the address type set to “IP4” followed by the address of the OITF. (e.g., c=IN IP4 <IP_ADDRESS>)
- The “b-line” shall contain the proposed bandwidth obtained by the OITF either by OPTIONS or during the service access phase. If the media stream is FEC protected and the OITF wishes to use one or more FEC streams, the bandwidth shall be the sum of the media stream bandwidth and the bandwidths of all the FEC stream to be used by the OITF. If the OITF cannot obtain the bandwidth, the b= attribute shall be set to a pre-configured value. If the media stream is retransmission protected and the OITF wishes to make use of retransmission, the bandwidth may be adjusted to take into account the maximum (peak) bandwidth the OITF can accept for receiving retransmissions. (e.g., b=AS:15000).
- A b=RR:<bandwidth-value> line indicating the bandwidth value (in kbit/s) that the OITF proposes to use for sending Receiver Reports (RR). If this value is set to zero by the OITF, then it means that the OITF cannot, or does not, wish to send Receiver Reports. This is the default setting, as explained in 10.2.2.2. Note that if the OITF sends RTCP Receiver Reports, then these can be used as keep-alive messages, as shown in 7.2.3.2.2.5. When the <proto> in the m-line is AVPF, the b=RR shall not be set to zero.
- An “a=” line with a “recvonly” (e.g., a=recvonly).

If a media stream is FEC protected, the OITF may include the following for each FEC protected stream:

- One or more m-line for the FEC streams indicated in the response to the OPTIONS request. The m-lines shall be set according to the returned response.

In case there are multiple media streams to be FEC protected, or a single media stream protected by multiple FEC streams, grouping line(s) shall be included for the purpose of associating FEC stream(s) with media stream(s), one for each media stream m-line that is associated to a FEC stream. The grouping line uses the “FEC” semantic as defined in IETF RFC 4756:

- a=group:FEC:<original stream id> <base FEC stream id> <enhancement FEC stream id>

The original stream id shall reflect the value held by the media description of media stream in the a=mid attribute. This implies that, when a grouping line is included, there shall be an additional media identification attribute within the m-line of the original media stream that is within the grouping line. The format for that attribute is:

- a=mid:<original stream id>

The base FEC stream id shall reflect the value held by the media description of the FEC stream (associated to the original stream) in the a=mid attribute.

Step 2: If the request is for a unicast content streaming session, the IG shall validate that the request includes all the mandatory SIP headers needed for the outgoing message, as per Table 16. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall send a SIP INVITE to the network, to request the initiation of a unicast session, and shall wait for the response to the request. The IG shall then return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the initiation request. The response includes a list of SIP headers as per Table 17, in addition to the normal HTTP headers as per IETF RFC 2616 IETF RFC 2616, and the same SDP answer body received by the IG, as described in 7.2.3.2.2.5.

Step 4: Upon receipt of a 200 OK response, the OITF shall send an HTTP Pending request to acknowledge the final response. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 18

HTTP request body: Empty

When parsing the b=RR:<bandwidth-value> line by the OITF: if the bandwidth value agreed is non-zero, then the OITF shall send RTCP RRs and shall not send RTSP keep-alive messages. If the bandwidth value received is zero, then the OITF shall not send RTCP RRs but instead it shall send RTSP keep-alive messages.

Table 16 – Supported HTTP extension headers in HNI-IGI INVITE request for unicast content streaming session setup (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request URI shall be set to the PSI (Public service Identifier) of the content service as follows: For CoD: OIPF_IPTV_CoD_service_*<domain name> Where: The wild card part (*) is a content instance identifier, constructed according to 4.3.2.2 in IEC 62776-3 when CoD content identifiers are delivered via the Content Guide. For DAE applications signalling CoD, the wild card part is constructed according to 8.2.2 in IEC 62776-5-1. The domain part (<domain name>) is the IPTV service provider domain name, obtained from the IPTV service provider discovery function.	IETF RFC 3261 INVITE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line INVITE" header	IETF RFC 3261
X-OITF-Contact Remarks: a) URI parameter shall be included and shall match what is sent in the contact header included in the registration request. b) Expires parameter should be included.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Supported	IETF RFC 3261 set to timer
X-OITF-session-Expires	IETF RFC 4028, IETF RFC 3556

Table 17 – Supported HTTP extension headers in the response to an HNI-IGI INVITE request for unicast content streaming session setup (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact The URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request.	IETF RFC 3261
X-OITF-session-Expires	IETF RFC 4028, IETF RFC 3556
X-OITF-Recv-Info To indicate willingness to receive the CoD-Bookmark Info Package, it shall be set to CoD-Bookmark or add CoD-Bookmark to the Info Package set; Or To indicate unwillingness, it shall be empty or remove CoD-Bookmark from the Info Package set. NOTE The CoD-Bookmark is defined in subclause 12 of 3GPP TS 26.237.	IETF RFC 6086

Table 18 – Supported HTTP extension headers in HNI-IGI ACK request for successful unicast content streaming session teardown (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message	IETF RFC 3261 ACK <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the “X-OITF-request-Line” of the initial request	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what has been inserted in the INVITE message. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261

6.3.2.1.3 Session termination

The OITF shall send the request for a unicast content streaming session termination using the following procedure:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 19

HTTP request body: empty

Step 2: The IG shall validate that the request includes all the mandatory SIP headers required for the outgoing message, as per Table 19. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall send a SIP BYE to the network, to request the termination of a unicast session, and shall wait for the response. The IG shall then return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the termination request. The response includes a list of SIP headers as per Table 20 in addition to the normal HTTP headers as per IETF RFC 2616.

Table 19 – Supported HTTP extension headers in HNI-IGI BYE request for unicast content streaming session teardown (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line	IETF RFC 3261 BYE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line" of the initial request	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

Table 20 – Supported HTTP extension headers in the response to an HNI-IGI BYE request for unicast content streaming session teardown (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.2.1.4 Session refresh

It is the responsibility of the OITF application to refresh the unicast content streaming session before the session expires. The IG shall consider a session terminated if it is not refreshed.

6.3.2.2 Protocol for streaming over UNIT-17

The use of the HTTP protocol on this reference point shall comply with IETF RFC 2616.

The content delivery function shall support the Range HTTP header in a GET request from the OITF to reduce unnecessary network usage by allowing partial retrieval for use in cases such as trick play. The OITF may pre-buffer the content in order to sustain play-out even when the HTTP transfer is stalled.

6.3.3 Forced play-out control with SIP session management – Protocol for forced play-out control over HNI-IGI

Given that Forced Play Out is completely supported in the network with no OITF impacts, the OITF shall initiate the request for unicast content streaming session using the procedure described in 6.3.2.1.2.

6.3.4 Content download

6.3.4.1 General

Content download is a service where IPTV content is downloaded to the optional internal storage system in the OITF. The OITF may play out the content while downloading. Trick play may be performed within the downloaded content depending on the content rights.

6.3.4.2 Protocol over UNIT-17

The use of the HTTP protocol on this reference point shall comply with IETF RFC 2616.

The content delivery function shall support the Range HTTP header in a GET request from the OITF to reduce unnecessary network usage by allowing partial retrieval.

6.3.5 Purchase of digital media service using SIP

6.3.5.1 General

A native application in the OITF may allow purchase of digital media related to a content item. Purchase of digital media is a service where the users want to buy and download additional digital media related to a content item in which they are interested. There are three steps to purchase digital media: advertising and retrieving the digital media related to the content item, the purchase request for selected digital media, and downloading the purchased digital media.

6.3.5.2 Retrieving the OIPF RelatedMaterial metadata for the content

The digital media related to the content is described through the OIPF RelatedMaterialType metadata which is extended from the TV-Anytime RelatedMaterial schema defined in ETSI TS 102 822-3-1. In order to provide advertising for the related material, the new element "Position_Size" is needed in the schema.

When the user pauses, OITF retrieves the OIPF RelatedMaterialType metadata of the content in order to advertise the related material. The new element "Position_Size" is used to define where promotional messages or media will appear on the user's screen and in what size.

The element Position_Size in RelatedMaterialType is shown in red in Figure 1.

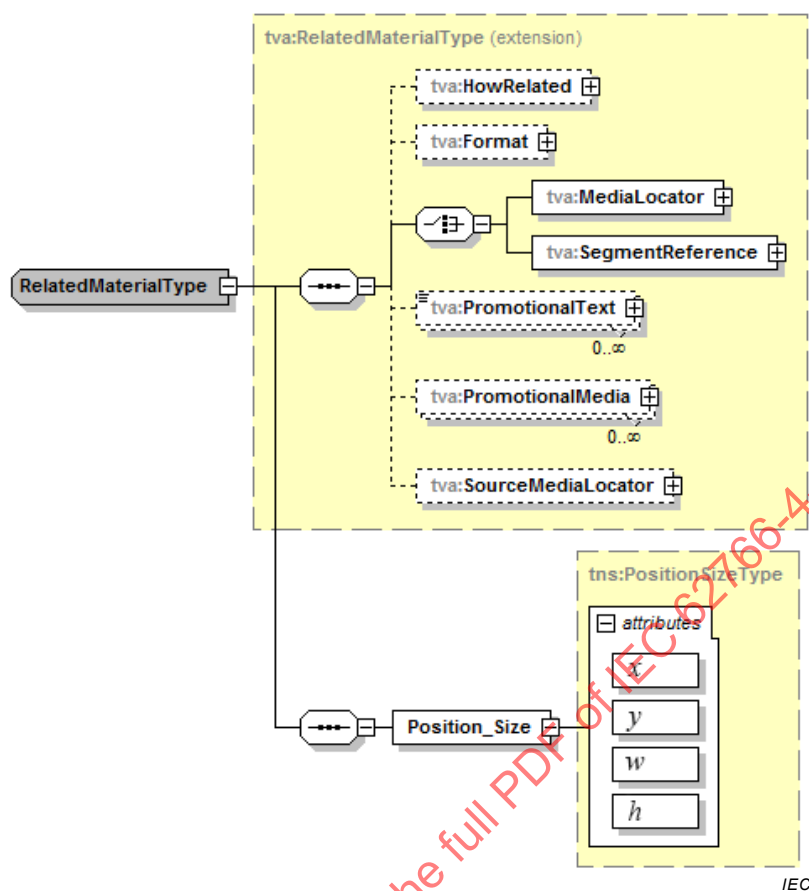
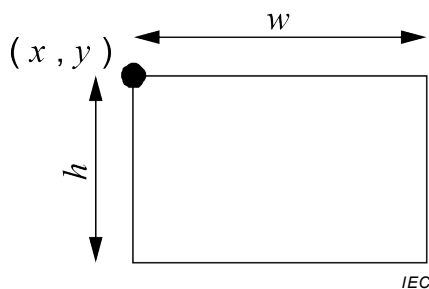


Figure 1 – Elements of OIPF extended RelatedMaterialType

Position_Size includes 4 numbers (x , y , w , h), where (x , y) means the top-left point of the advertisement, and w and h means the width and height of the advertisement, respectively. The position and size of each advertisement is decided and sent by IPTV Metadata control FE. The concept of constructing an advertising window using (x , y , w , h) in Position_Size element is shown in Figure 2.

Different resolution support: the Position_Size is set for the default resolution (for example, 800×600). If the OITF use another resolution (for example, $1\,024 \times 768$), the OITF scales the Position_Size to fit in with that resolution.



x , y : position
 w : width
 h : height

Figure 2 – Construction of advertising window using (x , y , w , h) in Position_Size element

6.3.5.3 Advertising digital media through the use of overlays

The OITF advertise digital media to users by overlaying the received PromotionalText and/or PromotionalMedia (e.g., a logo) of OIPF defined RelatedMaterialType and the original IPTV content, as shown in Figure 3.

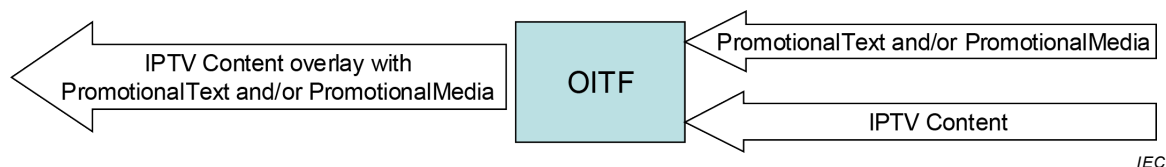


Figure 3 – Overlaying the content with advertisement for related media

6.3.5.4 Protocol over HNI-IGI – HTTP option

6.3.5.4.1 Purchase procedure for selected digital media related to the content

It is assumed that the OITF has an established session and has retrieved the necessary information to perform the purchase procedures as depicted in the steps below. It is also assumed that the network indicated its willingness to receive the Digital-Media-Purchase Info Package.

Step 1: The OITF sends HTTP request (with SelectedDigitalMediaURI, UserID) to IG for purchase selected digital media. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see per Table 21

HTTP request body: (As defined in 6.3.5.9, the “SelectedDigitalMediaURI” sets as the content of <MediaUri> in <MediaLocator> of OIPFRelatedMaterial, and “UserID” sets as <IMPU>).

Step 2: The IG shall validate that the request includes all the mandatory headers as per Table 21. The IG shall reject a request that is missing any mandatory SIP header with a non-200 HTTP OK response including the reason for rejection. Furthermore, it is assumed that the IPTV control FE indicated its willingness to receive the Digital-Media-Purchase Info Package at session setup time. Otherwise, the IG shall reject the request. The IG shall send a SIP INFO to the network including the Digital-Media-Purchase Info Package and shall wait for the response from the network.

Step 3: On receipt of a response message from network, the IG return shall return an HTTP 200 OK with the purchase result to OITF. The content of the HTTP response shall be as follows:

HTTP response header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see per Table 22

HTTP response body: purchase result. (It will be a simple string “success” if purchase request success, or an Unsigned 32bit “Result-Code” if purchase request failure).

Table 21 – Supported HTTP extension headers in the HNI-IGI SIP INFO request message for Digital-Media-Purchase (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INFO request shall be the well known PSI (Public service Identifier) of the Digital-Media-Purchase service: OIPF_IPTV_DMPurchase_service@<domain name>.	IETF RFC 3261 INFO <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF_Info-Package shall be set to "Digital Purchase"	IETF RFC 6086
X-OITF-Content-Type shall be set to the "application/vnd.oipf.purchase+xml"	IETF RFC 3261
X-OITF-Content-Disposition	IETF RFC 6086

Table 22 – Supported HTTP extension headers in the response message to an HNI-IGI INFO request message for Digital-Media-Purchase (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.5.4.2 Management of purchase request for digital media by network

The OITF shall comply with the following when it comes to network management for purchasing digital media:

- Step 1:** At any time, the IG can receive a SIP UPDATE from the network, and where the Recv-Info header is set to remove or re-instate support for reception of Digital-Media-Purchase Info Package to order the OITF to stop or start sending Digital-Media-Purchase requests. When a SIP UPDATE is received by the IG, the IG shall return an HTTP 200 OK response to the OITF. The response includes a list of SIP headers as per Table 23 in addition to the normal HTTP headers as per IETF RFC 2616.
- Step 2:** Once the OITF accepts the incoming SIP UPDATE, it shall send an HTTP HNI-IGI PENDING-IG request to acknowledge the receipt of the SIP UPDATE. The HTTP request includes a list of SIP headers as per Table 24 in addition to the normal HTTP headers as per IETF RFC 2616.

**Table 23 – List of HTTP extension headers
for an incoming SIP UPDATE (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INVITE shall match the contact URI included in the contact field of the SIP INVITE	IETF RFC 3261 UPDATE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 6086
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Recv-Info To indicate willingness to receive the Digital-Media-Purchase Info Package, it shall be set to "Digital-Media-Purchase" or add Digital-Purchase to the capability set; or To indicate unwillingness, it shall be empty or remove "Digital-Media-Purchase"	IETF RFC 6086

**Table 24 – Supported HTTP extension headers in the response message
to an incoming HNI-IGI SIP UPDATE request message (OITF→IG)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.5.5 Protocol over UNIP-1

Protocol for retrieving stored purchased digital media

If the OITF desires to retrieve the lists of purchased digital media stored in the network over UNIP-1, it shall send an XCAP GET request to the IPTV service profile FE. The IPTV service profile FE shall return all stored purchased digital media records in an HTTP 200 OK response.

NOTE The purchased digital media record can be updated only by the IPTV control FE, not the OITF.

6.3.5.6 Protocol over NPI-2 – Protocol for storing purchased digital media

Upon receipt of a request to store purchased digital media from the IPTV control FE, the IPTV applications FE shall send the XCAP PUT request to IPTV service profile FE to store a new purchased digital media record.

After updating the user's profile, the IPTV service profile FE shall return a response to IPTV applications FE, and then the IPTV applications FE shall return a response to IPTV control FE.

6.3.5.7 Protocol over UNIS-7

Use the following procedures when the OITF wants to retrieve OIPFRelatedMaterial of the content for advertisement.

NOTE The network initiated push to the OIPF RelatedMaterial is not specified.

Step 1: The OITF sends an HTTP request to IPTV metadata control FE to retrieving OIPFRelatedMaterial for the content for advertisement.

HTTP request header: Including the following:

- <list of HTTP headers> – as per Table 25

HTTP request body: It shall be a simple string OIPFRelatedMaterial request.

Step 2: The IPTV Metadata control FE returns the HTTP response to OITF with the OIPFRelatedMaterial for the content.

HTTP request header: Including the following:

- <list of HTTP headers> – as per Table 26

HTTP request body: As defined in 6.3.5.8.

Table 25 – Mandatory HTTP headers in the request message for advertisement of digital media (OITF→IPTV Metadata control FE)

HTTP header	Source of information for coding purposes
request-Line	IETF RFC 2616 POST <request URI> HTTP/1.1
Content-Length	IETF RFC 2616
Content-Type OPTIONAL. The default value of Content-Type is "text/plain; charset=us-ascii" as defined in IETF RFC 2045 IETF RFC 2045	IETF RFC 2616
Host	IETF RFC 2616

Table 26 – Mandatory HTTP headers in the response message for advertisement of digital media (IPTV Metadata control FE→OITF)

HTTP headers	Source of information for coding purposes
request-Line	IETF RFC 2616 HTTP/1.1 <Status-Code> <Reason-Phrase>
Content-Length	IETF RFC 2616
Content-Type shall be set to "application/vnd.oipf.OIPFRelatedMaterial+xml"	IETF RFC 2616

6.3.5.8 XML Schema for advertisement of digital media

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:service:RelatedMaterial:2011"
  xmlns:tns="urn:oipf:service:RelatedMaterial:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tva="urn:tva:metadata:2011">

  <xs:import namespace="urn:tva:metadata:2011"
    schemaLocation="imports/tva_metadata_3-1_v171.xsd"/>

  <xs:complexType name="RelatedMaterialType">
    <xs:complexContent>
      <xs:extension base="tva:RelatedMaterialType">
        <xs:sequence>
          <xs:element name="Position_Size" type="tns:PositionSizeType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PositionSizeType">
    <xs:attribute name="x" type="xs:unsignedShort" use="required"/>
    <xs:attribute name="y" type="xs:unsignedShort" use="required"/>
    <xs:attribute name="w" type="xs:unsignedShort" use="required"/>
    <xs:attribute name="h" type="xs:unsignedShort" use="required"/>
  </xs:complexType>
</xs:schema>
```

6.3.5.9 XML schema for purchase request of digital media

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:service:Purchaserequest:2010"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd" />

  <!-- "SelectedDigitalMediaURI" sets as the content of <MediaUri> in
    <MediaLocator> of RelatedMaterial -->
  <!-- "UserID" sets as <IMPU> -->
  <xs:complexType name="PurchaseDigitalMediarequestType">
    <xs:sequence>
      <xs:element name="SelectedDigitalMediaURI" type="xs:anyURI"/>
      <xs:element name="UserID" type="ct:UserIdType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

The SelectedDigitalMediaURI element represents the selected digital media that user wants to purchase.

The UserID element represents who made the Digital-Media-Purchase request.

6.3.6 Pay per view multicast content service with SIP session management protocol over HNI-IGI – HTTP option

6.3.6.1 General

When the OITF initiates, modifies or terminates a Pay-Per-View multicast content service, the OITF shall send HNI-IGI messages containing the appropriate method, mapped to HNI-IGI as described in 6.6.

The SIP-specific information in the related messages is described in 6.3.6. The SIP-specific information is mapped to the HNI-IGI protocol, as described in 6.6. In particular, the OITF creates HTTP headers for an HNI-IGI message by adding "X-OITF-" in front of the necessary SIP header names. In addition, OPTIONAL parameters may be included as defined in ETSI TS 124 503.

6.3.6.2 PPV service initiation without existing multicast content streaming session

PPV service initiation without an existing multicast content session is similar to the multicast content session initiation described in 6.3.1.2, with the following differences:

The media level of the SDP offer in the HTTP request body has the following additional requirements:

- The m-line(s) shall be set according to the mapping defined in Clause C.1 for the multicast content service to which the PPV service belongs.
- The c-line(s) shall be set according to the mapping defined in Clause C.1 for the multicast content service to which the PPV service belongs.
- An a=bc_service:BCserviceId line to indicate the multicast content service to which the PPV programme belongs shall be included.
- An a=bc_program:BCprogramId line to indicate the PPV programme shall be included.

When the OITF receives the response to the request, it shall examine the media parameters in the received SDP. The OITF shall restrict the multicast content services that it joins according to the parameter (the a=bc_service_package attribute) received from the IPTV control FE.

6.3.6.3 Switching from a PPV service to a multicast content service

To join a multicast content service outside the set of channels negotiated at session initiation, the OITF shall send a request to the IG for session modification. The OITF shall enforce a multicast content streaming session modification defined in 6.3.1.4.

To join a multicast content service inside the set of channels negotiated at session initiation, the OITF shall send an IGMP Leave request to stop watching the PPV service, and send an IGMP Join request to join the multicast content service. When FCC capability is available for the multicast content service, the OITF shall request a fast channel change as defined in Clause N.2. Further, when multicast RET service is offered, the OITF shall join the RET stream as defined in Clause N.1.

6.3.6.4 Switching to a PPV service from a multicast content service or another PPV service

The OITF shall send a request to the IG for session modification. The OITF shall generate a re-INVITE request, as defined in 6.3.6.2.

The OITF shall include an SDP offer in the session modification request. The format of this SDP offer shall be the same as for a session initiation. Note that session modification is only necessary if the PPV service is not included at session startup.

6.3.6.5 Session termination

When the OITF wants to terminate the session, the OITF shall generate a HTTP POST request as described in 6.3.1.5 for an originating OITF.

6.3.7 Parental control for content using SIP

6.3.7.1 General

Parental control relationship establishment, modification or deletion is out of scope of this document.

6.3.7.2 Protocol for what is on TV – OITF-initiated – HTTP option

6.3.7.2.1 Use-initiated subscription to acquire information related to watched content by another user

An IPTV end user having the parental control authority over another user can initiate subscription to acquire information related to the watched content by the user under his parental control. To initiate the request to receive the information, the OITF shall follow the following procedure:

- Step 1:** The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:
- HTTP request header including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 27
- HTTP request body: empty.
- Step 2:** The IG shall validate that the request includes all the mandatory SIP headers as per Table 27. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.
- Step 3:** The IG shall send a SIP SUBSCRIBE to the network, to subscribe to the Parental Control Watched Content event, and shall wait for the response to the subscription request. The IG shall return an HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the subscription request. The response shall include a list of SIP headers as per Table 28 in addition to the normal HTTP headers as per IETF RFC 2616.
- Step 4:** Following that, the OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any response.
- Step 5:** When a SIP NOTIFY is received by the IG, the IG shall return an HTTP 200 OK response to the OITF. The response shall include the list of SIP headers as per Table 29 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the SIP body received in the incoming NOTIFY. (See also 7.2.4.2.2.)
- Step 6:** Once the OITF accepts the incoming SIP NOTIFY, it shall send an HTTP HNI-IGI PENDING_IG request to the IG to send the SIP 200 OK response to the received SIP NOTIFY. The content of the HTTP request shall be as follows:
- HTTP request header: it includes the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 30
- HTTP request body: empty
- Step 7:** The IG shall send the SIP 200 OK response to the network and then shall return to Step 5 to handle any subsequent NOTIFY received from the network.

Table 27 – Supported HTTP extension headers in HNI-IGI SUBSCRIBE request for the parental control watched content event package

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI can be set in one of two ways: It can be set to the Public identity of the IPTV target end user under the parental control of the originator of the request in the From field or, it can be OPTIONALLY set to a PSI defined by the service provider. In either of the above cases, any SUBSCRIBE request to the Parental Control Watched Event shall be configured to be routed to the IPTV control FE (through iFC). This ensures that unauthorized requests are rejected by the network (unauthorized users may use the public identity of the target user instead of PSI in the request URI line)	IETF RFC 3261 and IETF RFC 3265 SUBSCRIBE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The To field is typically set to the public identity of the target user under the parental control of the originator of the request. If the To field is set to be identical to the From field, (i.e. the originator of the request) then the subscription is for all IPTV end users under the parental control of the originator of the request.	IETF RFC 3261
X-OITF-Event shall be set to the Event Package Parental-Control-Watched-Content	IETF RFC 3265 and parental control watched content (6.3.7.2.4)
X-OITF-Accept shall be set to "application/vnd.oipf.iptvparentalcontrol.whatsontv+xml"	IETF RFC 3265 and parental control watched content (6.3.7.2.4)
X-OITF-Contact Remarks: a) The URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request. b) Expires parameter should be included. c) Priority parameter should be included. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261

Table 28 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request for the parental control watched content event package

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 29 – Supported HTTP extension headers in the NOTIFY request containing changes in the watched programme (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP SUBSCRIBE	IETF RFC 3261 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3265 and parental control watched content (subclause 6.3.7.2.4)
X-OITF-Call-ID	IETF RFC 3261
X-OITF-Subscription-State	IETF RFC 3265
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.iptvparentalcontrol.whatsontv+xml"	IETF RFC 3265 and parental control watched content (6.3.7.2.4)
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 30 – Supported HTTP extension headers in the response to a NOTIFY request (OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

6.3.7.2.2 Terminating a subscription to the parental control watched content event package

This procedure shall be invoked at any time and/or prior to de-registering the user that created the subscription.

The procedure is the same as the procedure for initiating a subscription to the parental control watched content event, however in this case the X-OITF-Expires header in Table 27 shall be set to 0.

The IG shall consider a subscription terminated if is not refreshed.

6.3.7.2.3 Refreshing subscription to parental control watched event package

This procedure is the same as the procedure for initiating a subscription.

It is the responsibility of the application initiating the subscription procedure to refresh the subscription according to the refresh subscription timer information received in the response to the subscription request. Refreshing the subscription should be performed before the expiry of the refresh timer. A subscription that is not refreshed before the expiration of the refresh timer shall be terminated.

6.3.7.2.4 XML schema for parental control watched content

NOTE OITF contact information is mapped to DeviceId; OITF Call-ID is mapped to sessionId.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:WhatsOnTv:2011"
  xmlns:tns="urn:oipf:iptv:WhatsOnTv:2011"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:pss="urn:org:etsi:ngn:params:xml:ns:PssContentSwitchData"
  xmlns:mbms="urn:org:etsi:ngn:params:xml:ns:MbmsContentSwitchData"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd" />
  <xs:import
    namespace="urn:org:etsi:ngn:params:xml:ns:PssContentSwitchData"
    schemaLocation="imports/PssSwitchData.xsd"/>
  <xs:import
    namespace="urn:org:etsi:ngn:params:xml:ns:MbmsContentSwitchData"
    schemaLocation="imports/MbmsSwitchData.xsd"/>
  <xs:element
    type="tns:WhatsOnTvresponseType"
    name="WhatsOnTvresponse" />
  <xs:complexType name="WhatsOnTvresponseType">
    <xs:sequence>
      <xs:element name="User" type="tns:UserType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UserType">
    <xs:sequence>
      <xs:element name="name" type="ct:UserIdType"/>
      <xs:element name="content" type="tns:WatchedContentType"
        minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="WatchedContentType">
    <xs:choice>
```

```

        <xs:element name="PssWatchedContent"
type="tns:PssWatchedContentType"/>
        <xs:element name="MbmsWatchedContent"
type="tns:MbmsWatchedContentType"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="PssWatchedContentType">
    <xs:sequence>
        <xs:element ref="pss:PssSwitchData"/>
        <xs:element name="DeviceId" type="xs:string"/>
        <xs:element name="sessionId" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MbmsWatchedContentType">
    <xs:sequence>
        <xs:element ref="mbms:MbmsSwitchData"/>
        <xs:element name="DeviceId" type="xs:string"/>
        <xs:element name="sessionId" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

6.3.7.3 Protocol for parental control over HNI-IGI – HTTP option

6.3.7.3.1 Protocol for OITF originating a request for parental control

The user to be controlled is watching content and the controller finds out the information related to the watched content (SC service ID or content ID, ratings etc.) as described in 6.3.7.2. When the controller wants to block the content program, the OITF of the controller shall originate a request for Parental Control. The SIP-specific information in the related messages is described in 7.2.3.6. The SIP-specific information is mapped to the HNI-IGI protocol, as described in 6.6.

Step 1: The OITF shall send an HTTP POST request to the IG using the HNI-IGI functionality, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 31

HTTP request body: The OITF shall include a body associated with the appid "urn:oiptv:application:iptv-parental-control".

The message body shall include the parameters related to Parental Control as follows.

- PC-Command: the command for parental control, e.g. channel change, session teardown.
- PC-ChannelChangedTo: when the PC-Command is channel change, this parameter may be included. It indicates the new channel to change to.
- PC-ContentControlled: the identifier of the content being blocked by the controller. For scheduled content, it shall be the SC service ID. For content on demand, it shall be the content ID.

NOTE The detailed XML schema refers to 6.3.7.3.3.

The Content-Type of the message body shall be set to "application/vnd.oiptv.iptvparentalcontrol+xml" as described in Table 31 for X-OITF-Content-Type header.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the message as per Table 31. The IG shall reject a request that is missing any

mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

- Step 3:** The IG shall send a SIP MESSAGE to the network. When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate response) to the OITF to report the response to the SIP MESSAGE. The response includes a list of SIP headers as per Table 32 in addition to the normal HTTP headers as per IETF RFC 2616.

Table 31 – Supported HTTP extension headers in the MESSAGE request for parental control (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the MESSAGE request shall be set to the public identity of the controlled user.	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From The From in the MESSAGE request shall be set to the identity of the controller.	IETF RFC 3261
X-OITF-To The To in the MESSAGE request shall be set to the identity of the controlled user.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.iptvparentalcontrol+xml"	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

Table 32 – Supported HTTP extension headers in the response message to a MESSAGE request message for parental control (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.7.3.2 Protocol for OITF receiving a request for parental control

The following procedure is supported in the OITF of the controlled user for receiving a request for parental control.

The incoming message can be handled either by a native application in the OITF, or in a DAE application. The same HNI-IGI message format is used in either case.

- Step 1:** The IG receives an SIP MESSAGE for parental control

- Step 2:** The IG shall forward the SIP MESSAGE to the OITF as an HTTP response to a PENDING_IG request. The list of SIP headers to be included in the notification forward to the OITF shall be as per Table 33. The body of the SIP MESSAGE shall be included in the HTTP body.

Step 3: Upon receipt of the message, the OITF shall check the Content-Type in the “X-OITF-Content-Type” to determine that it is a parental control request. Then the OITF shall examine the parameters in the body, and initiate a request corresponding to the parameters in PC-Command, PC-ContentControlled. If the value in the PC-Command is Channel Change, the OITF shall leave the channel indicated in PC-ContentControlled and join a new channel which may be pre-configured or indicated in PC-ChannelChangedTo, if present, as described in 9.2.1. If the value in the PC-Command is session Teardown, the OITF shall initiate a request to terminate the session of the multicast content indicated in PC-ContentControlled, as described in 6.3.2.1.3.

Step 4: The OITF shall issue an HTTP POST request. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see per Table 31

HTTP request body: empty

Step 5: The IG shall forward the SIP 200 OK to the network.

Table 33 – Supported HTTP extension headers in the MESSAGE request for parental control (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI shall be set to the Public identity of the controlled user.	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From The From shall be set to the Public identity of the controller.	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request-URI in the “X-OITF-request-Line”.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to “application/vnd.oipf.iptvparentalcontrol+xml”	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

6.3.7.3.3 XML Schema for parental control

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:parentalcontrol:2011"
  xmlns:tns="urn:oipf:iptv:parentalcontrol:2011"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation xml:lang="en">
      Defines the command for parental control
      associated with the controller
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd"/>
  <xs:element name="IPTVParentalControl"
    type="tns:tIPTVParentalControl" />
  <xs:complexType name="tIPTVParentalControl">
    <xs:sequence>
      <xs:element name="PC-Command" type="tns:tPCCCommand" />
      <xs:element name="PC-ChannelChangedTo"
        type="tns:tPCChannelChangedTo"
        minOccurs="0" />
      <xs:element name="PC-ContentControlled" type="ct:ProgramIdType" >
        <xs:annotation><xs:documentation xml:lang="en">
          Identifier of the content being controlled by the controller.
          May be SC
          service ID or content ID.
        </xs:documentation></xs:annotation>
      </xs:element>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="tPCCCommand" final="list restriction">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Channel Change" />
      <xs:enumeration value="session Teardown" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tPCChannelChangedTo" final="list restriction">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Identifier of a new channel to which the controlled user shall
        change.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:minLength value="0" />
      <xs:maxLength value="16" />
    </xs:restriction>
  </xs:simpleType>

</xs:schema>

```

6.3.8 Network-based user notification services

6.3.8.1 General

User notification service refers to the family of services that includes a notification being sent to an IPTV end user. The notification can be sent either to an OITF or to a cellular device depending on user preference. There are several types of notification services. Subclause 6.3.8 deals with notification services related to broadcast reminders where the user can subscribe to be reminded, through a notification, before a specific broadcast starts. The actual notification can be sent anytime before the programme starts at the discretion of the IPTV SP. Notification services related to broadcast reminders involve interaction with the EPG for setting up a notification request. The notification services AS is an MMS Parlay-X web services AS.

6.3.8.2 Protocol over HNI-IGI – HTTP option Native HNI-IGI (IMS-based) notification request setup procedure

To initiate a request to setup a notification service, the OITF shall follow the following procedure:

Step 1: As a pre-requisite, it is assumed that the user is interacting with EPG where supported notification services are depicted to the end user. Once the user makes a selection, the OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 34

HTTP request body: See 6.3.8.7.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers as per Table 34. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall send a SIP MESSAGE to the network, to setup the request and shall wait for the response. The IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the setup request. The response shall include a list of SIP headers as per Table 35 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 4: Following that, the OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any response.

Table 34 – List of HTTP extension headers for an outgoing SIP MESSAGE for setting up a user notification request (OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the well-known PSI for the notification service	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.network-based-user-notification+xml"	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact The URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request.	IETF RFC 3261

Table 35 – List of HTTP extension headers for the response to an outgoing SIP MESSAGE for setting up a user notification request (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.8.3 Protocol over UNIS-6 for DAE-based notification management

6.3.8.3.1 DAE-based notification request setup

Upon receipt of a notification setup request from the OITF, the IPTV application shall validate the request against the appropriate schema. Upon successful validation, the IPTV application stores the notification request in the user IPTV profile using XCAP PUT request for that purpose.

6.3.8.3.2 DAE-based update of pending notifications

Update of pending notification requests is achieved through XML re-writing of the IPTV user service profile. It includes three steps; first, all the user pending notification requests are retrieved from the user service profile; next, the end user performs the necessary deletions/alterations; and, finally, the updated pending notification request is saved in the user service profile.

The IPTV application issues an XCAP GET to the user service profile and displays the result to the user.

Upon receipt of a pending notification update request from the OITF, the IPTV application shall authorize the request, update its internal state, and subsequently shall issue an XCAP PUT to the IPTV service profile to store the updated pending notification requests. The IPTV service profile shall acknowledge the successful storage of the updated pending notification requests in an HTTP 200 OK to the IPTV application, which in turn returns the response back to the OITF.

6.3.8.3.3 DAE-based procedure for retrieving pending notification requests

Upon receipt of a pending notification retrieval request from the OITF, the IPTV application shall first authorize the request and subsequently shall issue an XCAP request to the IPTV service profile to retrieve all stored pending notification requests for the subject user. Upon receipt by the IPTV application of the stored information, it shall return to the OITF in an HTTP 200 OK response the requested information.

6.3.8.4 Protocol over UNIP-1

6.3.8.4.1 Protocol for updating pending notification requests

If the OITF desires to update pending notification requests stored in the network over UNIP-1, it shall send an XCAP GET request to the IPTV service profile. The IPTV service profile shall return all stored pending notification requests in an HTTP 200 OK response. Once the user completes the modification process, the OITF shall send an XCAP PUT request to the IPTV service profile to update pending notification requests.

If the IPTV service profile detects a change in the pending notification requests, it shall inform the IPTV control FE, which in turn notifies the IPTV application. The IPTV control FE shall wait for the response before sending an HTTP 200 OK back to the OITF (or an appropriate error message if applicable).

6.3.8.4.2 Protocol for retrieving stored pending notification requests

An OITF that desires to retrieve pending notification requests over UNIP-1, shall send an XCAP GET request to the IPTV service profile. The IPTV service profile shall return all pending notification requests in an HTTP 200 OK response.

6.3.8.5 Protocol over NPI-2

6.3.8.5.1 Procedure for notification request setup

Upon receipt of a store request for user notification from the IPTV control FE, the IPTV application shall validate the request against the appropriate schema.

Upon successful validation, the IPTV application shall update its internal state and shall store the notification request in the user IPTV profile using XCAP PUT request for that purpose prior to returning a response.

6.3.8.5.2 Procedure for updating (deleting) a pending notification

Upon receipt of an update (delete request) for a pending user notification request from the IPTV control FE, the IPTV application shall validate the request against its internal state and shall update its internal state accordingly.

Following that, the IPTV application shall acknowledge the update (deletion) or send an appropriate error response if applicable.

6.3.8.6 Protocol over NPI-36

Incoming request to the notification AS for delivery notification to an IPTV user

Upon receipt by the notification services AS (MMS Parlay X web services AS) for an HTTP POST request for sending a delivery notification using the IMPagerMode, the notification services AS shall send a corresponding SIP MESSAGE to the messaging AS, and shall wait for the SIP 200 OK (or other responses) before returning the corresponding HTTP response.

Upon receipt by the IPTV application of the HTTP 200 OK, it shall proceed to update its internal state and shall update as well the IPTV service profile to reflect the updated list of pending notification requests.

6.3.8.7 XML schema for notification request for broadcast reminder

Broadcast reminders related to notification are sent to the end user before a broadcast starts. It typically involves interaction with the EPG.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:reminder:2011"
  xmlns:tns="urn:oipf:iptv:reminder:2011"
  xmlns:tva="urn:tva:metadata:2011"
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd"/>
  <xs:import namespace="urn:tva:metadata:2011"
    schemaLocation="imports/tva_metadata_3-1_v171.xsd"/>
  <xs:element name="ReminderList">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Reminder" type="tns:ReminderType" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ReminderType">
    <xs:sequence>
      <xs:element name="Creator" type="ct:UserIdType"/>
      <xs:element name="Created" type="xs:dateTime"/>
      <xs:choice>
        <xs:element name="ProgramIdentifier" type="ct:ProgramIdType"/>
        <xs:element name="SeriesIdentifier" type="tva:EpisodeOfType"/>
      </xs:choice>
      <xs:element name="AnnouncementTime" type="xs:duration"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

6.3.9 Content bookmarking

6.3.9.1 General

Content bookmarking is a feature that allows a user to store in the network a point in time for a scheduled content or content on demand session where later it can be retrieved and where viewing can resume from that point on.

For a scheduled content session, bookmarking essentially represents a mark in a file stored in the network for the scheduled content. As such, it is a pre-requisite that the scheduled content

be stored in the network for any bookmarking to be available for a scheduled content session. The stored bookmarking hence will be a pointer in a file storing the scheduled content.

6.3.9.2 Protocol over HNI-IGI – HTTP option

6.3.9.2.1 IMS-based content bookmark creation request

The OITF shall follow the following procedure for a content bookmark creation request:

Step 1: If the OITF decides to store a content bookmark for watched content during a multicast or unicast content streaming session, and if it is permitted by the network to do so, the OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 38

HTTP request body: shall include the Content Bookmark based on the XML schema as per 6.3.9.6.

Note that the SIP INFO including the CoD-Bookmark Info Package according to subclause 12 of 3GPP TS 26.237 is sent only within the context of a multicast or unicast content streaming session.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the process as per Table 38. The IG shall send a SIP INFO including the CoD-Bookmark Info Package to the network and shall wait for the response to the request. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: On receipt of the response from the network the IG shall return a HTTP 200 OK response (or other appropriate received responses) to the OITF to report the response to the INFO request. The response shall include a list of SIP headers as per Table 39 in addition to the normal HTTP headers as per IETF RFC 2616.

If the OITF desires to provide the user with the Replay URL, it can optionally initiate the proper procedures for that purpose. This is an option that is available to the OITF.

6.3.9.2.2 Management of content bookmarking creation request by the network

The OITF shall comply with the following when it comes to network management for storing a content bookmark in the network, for either a multicast or unicast content streaming (scheduled content or a CoD):

Step 1: At any time, the IG can receive a SIP UPDATE from the network, and where the Recv-Info header is set to remove or re-instate support for reception of CoD-Bookmark Info Package according to subclause 12 of 3GPP TS 26.237 to order the OITF to stop or start sending content bookmark creation requests. When a SIP UPDATE is received by the IG, the IG shall return an HTTP 200 OK response to the OITF. The response includes a list of SIP headers as per Table 40 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 2: Once the OITF accepts the incoming SIP UPDATE, it shall send an HTTP HNI-IGI PENDING-IG request to acknowledge the receipt of the SIP UPDATE. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 41

HTTP request body: empty

6.3.9.2.3 Content-related bookmark retrieval over HNI-IGI

The procedure for retrieval of content-related bookmarks is similar to the procedure defined in 6.3.2.1.2 with the addition that the OITF has indicated its willingness to receive the CoD-Bookmark Info Package according to Clause 12 of 3GPP TS 26.237 at session initiation. Assuming that is the case, the following additional steps are added to the existing procedure:

Step 1: The IG shall wait for the SIP INFO with the Bookmark list from the network. The IG shall then return a HTTP 200 OK response to the OITF to report the Bookmark list. The response includes a list of SIP headers as per Table 36, in addition to the normal HTTP headers as per IETF RFC 2616, and XML (see 6.3.9.6) in the SIP INFO body received by the IG, as described in 7.2.3.8.3.1. The IG then returns a SIP 200 OK to the network.

Step 2: Upon receipt of a 200 OK response, the OITF shall send an HTTP Pending request to acknowledge the final response. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 37.

HTTP request body: Empty

After the OITF retrieves the content-related bookmark lists, the OITF displays the bookmark list to the user, and the user selects the bookmark from which she wishes to start viewing the content.

**Table 36 – Supported HTTP extension headers
for an incoming SIP INFO (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line The request-URI in the SIP INFO request shall be the contact included in the INVITE request message, that is the user ID (IMPU).	IETF RFC 3261 INFO <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/3gpp-ims-pss-mbms-bookmark+xml" that corresponds to Annex J of 3GPP TS 26.237	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Recv-Info To indicate willingness to receive the CoD-Bookmark Info Package according to subclause 12 of 3GPP TS 26.237, it shall be set to CoD-Bookmark or add CoD-Bookmark to the capability set; or To indicate unwillingness, it shall be empty or remove CoD-Bookmark	IETF RFC 6086

Table 37 – Supported HTTP extension headers in the response message to an incoming HNI-IGI SIP INFO (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the ACK request shall be the contact included in the incoming SIP INFO message	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line" of the INVITE request	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what has been inserted in the INVITE message. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261

6.3.9.3 Protocol over UNIS-6 for DAE-based content bookmark management

6.3.9.3.1 General

UNIS-6 may be used for the unicast transport of HTML ECMAScript documents between the OITF DAE function and the IPTV application FE for DAE-based subscription profile management. In this case, the IPTV application FE acts as a front-end to the IPTV service profile FE. When the HTTP request for bookmark management is received from OITF, the IPTV application FE manipulates the IPTV service profile FE.

6.3.9.3.2 DAE-based content bookmark creation request

Upon receipt of an HTTP request for a content bookmark creation request from the OITF, the bookmark IPTV application shall authorize the request and shall verify if the bookmark is for scheduled content or CoD.

If the content bookmark is for a scheduled content, the bookmark IPTV application verifies first if the content is available for bookmarking, i.e. stored in the network. If available for bookmarking, the bookmark IPTV application, after performing the necessary adaptation, stores the bookmark information in the IPTV service profile using XCAP PUT request for that purpose.

For bookmarks related to CoD, the bookmark information is stored in the user IPTV service profile without any additional verification by the IPTV application.

6.3.9.3.3 DAE-based bookmark retrieval request

Upon receipt of a content bookmark retrieval request from the OITF, the bookmark IPTV application shall first authorize the request and subsequently shall issue an XCAP GET request to the IPTV service profile to retrieve all stored content bookmarks for the user.

6.3.9.3.4 DAE-based content bookmark update over UNIS-6

Content Bookmark update is essentially achieved through XML re-writing of the content bookmarks in the IPTV user service profile. It includes three steps; first all the content bookmarks are retrieved from the IPTV service profile; next the end user performs locally the

necessary update (deletions/alterations); and finally the updated content bookmark is saved in the IPTV service profile.

The bookmark IPTV application shall authorize the request and subsequently shall issue an XCAP GET to the IPTV service profile and display the results to the user.

Upon receipt of a content bookmark request from the OITF, the bookmark IPTV application shall authorize the request and subsequently shall issue an XCAP PUT to the IPTV service profile to store the updated content bookmarks. The IPTV service profile shall acknowledge the successful storage of the updated content bookmarks in an HTTP 200 OK to the bookmark IPTV application which in turn returns the response back to the OITF.

Note that XCAP supports several approaches for retrieving/storing bookmark information within the user profile. It is an implementation issue as to which approach the IPTV application implements.

6.3.9.4 Protocol over NPI-2

Protocol for storing content bookmarks

Upon receipt of a bookmark store request from the IPTV control FE, the bookmark IPTV application shall verify if the bookmark is for scheduled content or CoD.

If the content bookmark is for a scheduled content, the bookmark IPTV application verifies first if the content is available for bookmarking, i.e. stored in the network. If available for bookmarking, the bookmark IPTV application, after performing the necessary adaptation, stores the content bookmark information in the user IPTV service profile using XCAP PUT request for that purpose.

For bookmarks related to CoD, the content bookmark information is stored in the user IPTV service profile without any additional verification by the bookmark IPTV application.

Following the completion of the XCAP transaction, a response is sent to the IPTV control server FE.

6.3.9.5 Protocol over UNIP-1

6.3.9.5.1 Protocol for retrieving stored content bookmarks

If the OITF desires to retrieve content bookmarks stored in the network over UNIP-1, it shall send an XCAP GET request to the IPTV service profile. The IPTV service profile shall return all stored bookmarks in an HTTP 200 OK response.

6.3.9.5.2 Protocol for updating content bookmarks

If the OITF desires to update a content bookmark(s) stored in the network over UNIP-1, it shall send an XCAP GET request to the IPTV service profile. The IPTV service profile shall return all stored bookmarks in an HTTP 200 OK response. Once the user completes the update process, the OITF shall send an XCAP PUT request to the IPTV service profile to update the content bookmarks. The IPTV service profile shall acknowledge the successful storage of the updated content bookmarks in an HTTP 200 OK response.

Table 38 – Supported HTTP extension headers in the HNI-IGI SIP INFO request message for content bookmark (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INFO request shall be the well-known PSI (Public service Identifier) of the bookmarked content service. For a Scheduled service: OIPF_IPTV_SC_service@<domain name>. For a Content on Demand: OIPF_IPTV_CoD_service_*@<domain name>.	IETF RFC 3261 INFO <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what is returned in the Contact header included in the response to the registration process. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
X-OITF-Call-ID shall match what was used in the SIP INVITE	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Info-Package shall be set to CoD-Bookmark according to subclause 12 of 3GPP TS 26.237.	IETF RFC 6086
X-OITF-Content-Type shall be set to "application/3gpp-ims-pss-mbms-bookmark+xml" that corresponds to Annex J of 3GPP TS 26.237	IETF RFC 6086, 6.3.9.6
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Content-Disposition	IETF RFC 6086

Table 39 – Supported HTTP extension headers in the response message to an HNI-IGI INFO request message for content bookmark (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

Table 40 – List of HTTP extension headers for an incoming SIP UPDATE (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the INVITE shall match the contact URI included in the contact filed of the SIP INVITE	IETF RFC 3261 UPDATE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 6086
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Recv-Info To indicate willingness to receive the CoD-Bookmark Info Package according to subclause 12 of 3GPP TS 26.237, it shall be set to CoD-Bookmark or add CoD-Bookmark to the capability set; or To indicate unwillingness, it shall be empty or remove CoD-Bookmark	IETF RFC 6086

Table 41 – Supported HTTP extension headers in the response message to an incoming HNI-IGI SIP UPDATE request message (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.3.9.6 XML schema for content bookmarking

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:bookmark:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn:oipf:iptv:bookmark:2011"
  xmlns:bm3gpp="urn:3gpp:bookmark:2009:IMS-PSS-MBMS"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:3gpp:bookmark:2009:IMS-PSS-MBMS"
    schemaLocation="imports/3gpp-bookmark-2009-IMS-PSS-MBMS.xsd"/>

  <xs:complexType name="BookmarkType">
    <xs:complexContent>
      <xs:extension base="bm3gpp:BookmarkType">
        <xs:sequence>
          <xs:element name="Location" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```
</xs:complexContent>
</xs:complexType>
</xs:schema>
```

The following elements shall be provisioned by the entity submitting the content bookmark for storage:

- Creator. This element is set to the IMS Public Identity (IMPU).
- Created.
- Comment. This element represents any comment chosen by the user.
- Tag. This element represents any categorization chosen by the user.
- Rank. This element represents the user favourite rating for the bookmark.
- Sharing. If set to “true”, the bookmark can be shared with others.
- Retrieval count shall be set to 0 and incremented by the service provider when the bookmark is retrieved.

The following elements and attributes shall be provisioned by the service provider before storing the content bookmark in the user service profile:

- Expires;
- id.

The following elements shall be conditionally provisioned by the OITF or the service provider as described below:

- ProgramId and ProgramType. This reflects the identifier of the content being bookmarked. This information is typically submitted by the OITF.
 - For scheduled content, the CRID for the programme currently being watched shall be included in this ProgramId element and the ProgramType element shall be set to “SC”;
 - for content on demand the CRID for the content shall be included in the ProgramId element and the ProgramType element shall be set to “CoD”.
 - The network shall verify provisioned information in case of scheduled content and content on demand, and shall ensure that the information provided is accurate. If the provisioned information is inaccurate or missing, it shall be corrected/completed respectively by the network, where applicable. Furthermore, for scheduled content, the network shall ensure that the programme is stored in the network before the request is accepted.
- Location. This item includes the bookmark URL. The URL shall allow the user to stream the content at a later time from the bookmarked position. The information shall be created by the network, and included in the element before storing the information in the IPTV service profile
- Offset. This element includes the bookmark position in the form of an offset from the beginning of the stream.
 - For content on demand, the offset shall be included by the OITF and shall be verified before storage, if possible.
 - For scheduled content, the OITF shall set the offset to PT0S. The network shall first ensure that the content is recorded in the network. The network shall consult with the EPG and the time of the incoming request to calculate the offset prior to storage in the network. The calculated stored offset shall take into account the network storage aspects into consideration to ensure accurate mapping when streaming is later activated through the bookmark URL.
- Retrieval time shall be updated with a new entry every time the content bookmark is retrieved (note that retrieval does not imply the content bookmark is used).

6.3.10 Local PVR service using SIP

6.3.10.1 Protocol over HNI-IGI – HTTP option

The OITF initiates the request for PVR service capture request using the following procedure:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 42

HTTP request body: The OITF shall include a body associated with the appid “urn:oipf:service:PVR:2011”.

- The message body shall include the parameters related to ctCapturerequest Type choice setup as follows.
 - requestType: indicates type of service request (i.e. SetUpRecording, CancelRecording, UpdateRecording, RetrievalRecording.);
 - ProgramId: identifies the content or programme to be recorded in the OITF;
 - BCserviceId: identifies scheduled content channel from which the programme is to be recorded in the OITF;
 - ProgramStartTime: indicates the time to start the recording that the OITF requests;
 - ProgramDuration: indicates the time duration of the recording that the OITF requests;
 - StorageRecMode: indicates the location where requested content will be recorded (i.e. Local, Network); in case of local PVR service, the StorageRecMode shall be “Local”;
 - TargetDeviceId: identifies the target local PVR (OITF) where contents are recorded. This parameter shall be shown as syntax of the sip.instance feature tag or GRUU.

According to the requestType, the OITF shall initiate a request for recording order setup, recording order cancel, recording order update, recording order retrieval.

The Content-Type of the message body shall be set to “application/vnd.oipf.pvr+xml” as described in Table 42 for X-OITF-Content-Type header.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the process as per Table 42. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall send a SIP MESSAGE to the network to initiate LPVR as requested by the OITF, and shall wait for the response to the request. The IG shall return HTTP 200 OK response (or other appropriate response) to the OITF to report the response to the PVR service capture request. The response shall include a list of SIP header as per Table 43 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 4: The OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1) and shall wait for any response.

Step 5: When a SIP MESSAGE is received by the IG, the IG shall return an HTTP 200 OK response to the OITF. The response shall include the list of SIP header as per Table 45 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the SIP body received in the incoming MESSAGE message. The content of the HTTP response shall be as follows:

HTTP response header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 45

HTTP response body: The OITF shall include a body associated with the appid “urn:oipf:service:PVR:2011”.

- The message body shall include the parameters related to ctLPVRRecordrequest Type choice setup as follows.
 - requestType: indicates type of service request (i.e. SetUpRecording, CancelRecording, UpdateRecording, RetrievalRecording).
 - ProgramId: identifies the content or programme to be recorded in the OITF.
 - BCserviceId: identifies scheduled content channel from which the programme is to be recorded in the OITF.
 - ProgramStartTime: indicates the time to start the recording that the OITF requests.
 - ProgramDuration: indicates the time duration of the recording that the OITF requests.
 - StorageRequirement: indicates the required storage of Local PVR for recording requested content.

Note that at any time, the IG can receive such a message from the network to perform a new request type on a pending recording request (e.g., recording order cancel, recording order update).

The Content-Type of the message body shall be set to “application/vnd.oipf.pvr+xml” as described in Table 42 for X-OITF-Content-Type header.

Step 6: Once the OITF accepts the incoming request, the OITF shall send an HTTP POST request (refer to 6.6.1) to the IG to convey the SIP response. The HTTP request shall include the SIP headers as per Table 43 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 7: When time to order recording of a requested content is up, the OITF shall initiate the multicast content streaming session setup. The OITF shall follow the steps defined by 6.3.1.2.

Table 42 – List of HTTP extension headers for PVR service capture request message (OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line The request URI shall be set to the well-known PSI for the PVR service.	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.pvr+xml"	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact The URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request.	IETF RFC 3261

Table 43 – List of HTTP extension headers for the response to PVR service capture request and PVR record request message (IG→OITF and OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

Table 44 – List of HTTP extension headers for the PVR record request (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line The request URI shall be set to the Public identity of the target of the message	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.pvr+xml"	IETF RFC 3261
X-OITF-Accept-Contact	Set to the SIP Instance of the TargetDevice This parameter includes required and explicated as IETF RFC 3841
X-OITF-Content-Length	IETF RFC 3261

6.3.10.2 XML schema for PVR service

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:service:PVR:2011"
  xmlns:tns="urn:oipf:service:PVR:2011"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ueprofile="urn:oipf:iptv:UEprofile:2010"
  xmlns:iptvprofile="urn:oipf:iptv:IPTVprofile:2011"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd" />
  <xs:import namespace="urn:oipf:iptv:UEprofile:2010"
    schemaLocation="iptv-UEprofile.xsd"/>
  <xs:import namespace="urn:oipf:iptv:IPTVprofile:2011"
    schemaLocation="iptv-IPTVprofile.xsd"/>
  <xs:element name="PVR">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="serviceType" type="tns:ctserviceType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ctserviceType">
    <xs:choice>
      <xs:element name="Capturerequest" type="tns:ctCapturerequest"
        maxOccurs="unbounded"/>
      <xs:element name="LPVRRecordrequest"
        type="tns:ctLPVRRecordrequest"
        maxOccurs="unbounded"/>
      <xs:element name="NPVRRecordrequest"
        type="tns:ctNPVRRecordrequest"
        maxOccurs="unbounded"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ctCapturerequest"><!--LPVR Capture
request/response-->
```

```

<xs:sequence>
  <xs:element name="requestType" type="tns:strequestType"/>
  <xs:element name="ProgramID" type="ct:ProgramIdType"/>
  <xs:element name="BCserviceID" type="iptvprofile:tBCserviceID"/>
  <xs:element name="ProgramStartTime" type="xs:dateTime"/>
  <xs:element name="ProgramDuration" type="xs:duration"/>
  <xs:element name="StorageRecMode" type="tns:stStorageRecMode"/>
  <xs:element name="TargetDeviceID" type="ueprofile:tUEID"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ctLPVRRecordrequest"><!--LPVR          Record
request/response-->
  <xs:sequence>
    <xs:element name="requestType" type="tns:strequestType"/>
    <xs:element name="ProgramID" type="ct:ProgramIdType"/>
    <xs:element name="BCserviceID" type="iptvprofile:tBCserviceID"/>
    <xs:element name="ProgramStartTime" type="xs:dateTime"/>
    <xs:element name="ProgramDuration" type="xs:duration"/>
    <xs:element name="StorageRequirement"
type="tns:ctStorageRequirement"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ctNPVRRecordrequest"><!--NPVR          Record
request/response-->
  <xs:sequence>
    <xs:element name="requestType" type="tns:strequestType"/>
    <xs:element name="ProgramID" type="ct:ProgramIdType"/>
    <xs:element name="BCserviceID" type="iptvprofile:tBCserviceID"/>
    <xs:element name="ProgramStartTime" type="xs:dateTime"/>
    <xs:element name="ProgramDuration" type="xs:duration"/>
    <xs:element name="StorageRequirement"
type="tns:ctStorageRequirement"
minOccurs="0"/>
    <xs:element name="requestStatus" type="tns:trequestStatus"
minOccurs="0"/>
    <xs:element name="CRID" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="strequestType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SetUp"/>
    <xs:enumeration value="Cancel"/>
    <xs:enumeration value="Update"/>
    <xs:enumeration value="Retrieval"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="stStorageRecMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Local"/>
    <xs:enumeration value="Network"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ctStorageRequirement">
  <xs:simpleContent>
    <xs:extension base="xs:int">
      <xs:attribute name="unit" type="xs:string" use="optional"
default="KB"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="trequestStatus">

```

```

<xs:restriction base="xs:string">
  <xs:enumeration value="Recording Scheduled"/>
  <xs:enumeration value="Recording Started"/>
  <xs:enumeration value="Recording Completed"/>
  <xs:enumeration value="Recording Deleted"/>
  <xs:enumeration value="Recording Failed"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>

```

6.3.11 Network PVR (nPVR) using SIP

6.3.11.1 Protocol over HNI-IGI – HTTP option

When the OITF initiates the request for network PVR service request, the procedure defined in 6.3.10 shall apply, with following additional constraints.

In step 1, the HTTP request body shall be set so that in the PVR request message ctNPVRRecordrequest shall be the selected choice and shall be set as follows:

- requestType: indicates type of service request;
- ProgramId: identifies the content or programme to be recorded in the OITF;
- BCserviceld: identifies the scheduled content channel from which the programme is to be recorded in the OITF;
- ProgramStartTime: indicates the time to start the recording that the OITF requests;
- ProgramDuration: indicates the time duration of the recording that the OITF requests;
- StorageRecMode: shall be set to "Network".

Steps 5-6 are replaced by the following steps:

Step 5: When a SIP MESSAGE is received by the IG due to the failure of a submitted request or to report the outcome of a pending recording request, the IG shall return an HTTP 200 OK response to the OITF. The response shall include the list of SIP headers as per Table 45. The body of the HTTP response shall include the SIP body received in the incoming SIP MESSAGE. The content of the HTTP response shall be as follows:

HTTP response header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – see Table 45

HTTP response body: The actual content included in the SIP MESSAGE body.

Step 6: The OITF shall extract the pertinent information from the body and shall send a SIP 200 OK response to the network in an HTTP POST request.

Table 45 – List of HTTP extension headers for the PVR record request (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line The request URI shall be set to the Public identity of the target of the message	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.pvrresult+xml"	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

6.3.11.2 XML schema for nPVR recording result

This subclause specifies the XML schema of the record status report which the CDF send to the cluster controller FE.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:service:PVR:report:2010"
  xmlns:tns="urn:oipf:service:PVR:report:2010"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tva="urn:tva:metadata:2011"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:tva:metadata:2011"
    schemaLocation="imports/tva_metadata_3-1_v171.xsd"/>
  <xs:element name="PVRResult">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RecordResult" type="tns:tRecordingResult"/>
        <xs:element name="NPVRLocation" type="tns:tNPVRLocator"/>
        <xs:element name="SpareStorage" type="xs:double"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="tRecordingResult">
    <xs:sequence>
      <xs:element name="RecordingStatus" type="tns:tRecordStatus"/>
    </xs:sequence>
    <xs:attribute name="ErrorCode" type="xs:string" use="optional"/>
  </xs:complexType>
  <xs:simpleType name="tRecordStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Completed"/>
      <xs:enumeration value="Partial"/>
      <xs:enumeration value="Error"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="tNPVRLocator">
    <xs:choice>
      <xs:element name="CRID" type="tva:CRIDType"/>
      <xs:element name="URL" type="xs:anyURI"/>
    </xs:choice>
  </xs:complexType>
```

</xs:schema>

If recording is successful, the ErrorCode element attribute shall not be present. If recording failed (i.e., RecordResult element contains “Error”), any information in the NPVRLocator element shall be discarded by the OITF.

6.3.12 Personalised channel

6.3.12.1 Procedure for network-centric personalized channel (unicast only)

6.3.12.1.1 General

These procedures assume that the personalised channel content will be served by a single CDF, regardless of where the content is sourced.

6.3.12.1.2 PCh profile configuration

When the user wants to create a new personalised channel, the OITF of the user shall originate an HTTP GET request towards the IPTV application via UNIS-6 reference point. The request shall carry the user ID.

When receiving the HTTP GET request for PCh configuration, the IPTV application shall authorize and verify the request, and then send XCAP GET request to the IPTV service profile and retrieve the user’s IPTV service profile. The XCAP GET request shall be delivered on NPI-17 and include the intended user ID.

After the request is authorized, the IPTV application shall contact the IPTV Metadata Control via NPI-33 for searching and filtering of the content metadata, and generate the personalised content guide based on the user’s personal setting (e.g. preference, location etc). The IPTV application shall also create a valid Personalised Channel Identifier (PChId) for the generated PCh information.

The IPTV application then responds 200 OK to the OITF via UNIS-6, carrying the generated PChId and associated PCh information, e.g., PCh item IDs (SC service ID or COD content ID), related time schedule, etc.

If the OITF desires to modify the PCh information that has been created, it shall send an HTTP PUT request through UNIS-6 to the IPTV application, carrying the intended user ID and the PCh information for updating. The IPTV application shall update the PCh information towards the IPTV service profile FE through XCAP PUT over NPI-17, and then responds the OITF with a 200 OK.

6.3.12.1.3 PCh service provision

6.3.12.1.3.1 PCh session initiation over HNI-IGI

When the user wants to watch the programs scheduled in their PCh, the OITF of the user shall originate a request for PCh service setup. The procedure is similar to the procedure described in 6.3.2.1.2, with the exception that the X-OITF-request-Line extension header delivered over HNI-IGI (Table 16) shall include the Personalised Channel Identifier (PChId), e.g. IPTV_PCH_service_PChId@<domain name>, where:

- The Personalised Channel Identifier (PChId) is retrieved from the IPTV application in the PCh configuration procedure, 6.3.12.1.2.
- The domain part (<domain name>) is the IPTV service provider domain name, obtained from the IPTV service provider discovery function.

6.3.12.1.3.2 PCh session termination over HNI-IGI

The OITF shall send the request for a Personalised Channel session termination using the procedure as described in 6.3.2.1.3.

6.3.12.2 OITF-centric personalised channel

When the user wants to create a new Personalised Channel, the user's OITF may originate an HTTP GET request towards the IPTV application via the UNIS-6 reference point. The request shall carry the user ID.

When receiving the HTTP GET request for PCh configuration, the IPTV application shall authorize and verify the request, and then send an XCAP GET request to the IPTV service profile and retrieve the user's IPTV service profile. The XCAP GET request shall be delivered on the NPI-17 interface and includes the intended user ID.

The IPTV application then responds with a 200 OK message to the OITF via UNIS-6, carrying the user profile that can be used for generating the Personalised Channel Guide at the OITF.

The OITF should ensure that the Personalized Channel does not include a time gap. When the OITF detects a time gap between adjacent content items in the PCh schedule, it determines if the length of the time gap is longer than a specified threshold (e.g. user defined), and may insert padding content into the time gap.

The padding content can be obtained from a PCh compatible source, e.g. PVR or other Home Network device as shown in the Figure 30 of IEC 62766-4-2:2017.

When the OITF detects an overlap between adjacent content items in the PCh schedule, the OITF decides the location of the PVR (LPVR or nPVR) used to record the overlapped contents based on either a pre-configured policy or the capability of OITF or network (ex, storage, bandwidth, etc).

The messaging and procedures for recording the overlapped content item on an LPVR shall be as specified in 6.3.10.

The messaging and procedures for recording the overlapped content item on an nPVR shall be as specified in 6.3.11.

After the overlapped content item has finished recording and the time gap has been filled, the OITF should update the Personalised Channel Guide with the revised content access information.

The OITF sets up the proper session for content delivery or plays the locally stored content according to the already configured Personalized Channel Guide.

6.3.13 Session transfer with SIP session management

6.3.13.1 Protocol over HNI-IGI – HTTP option

6.3.13.1.1 Generic session transfer procedures

6.3.13.1.1.1 General

The procedures in this subclause are generic in nature and apply equally to the various modes of session transfer.

6.3.13.1.1.2 Transferee unicast session streaming session initiation associated with a session transfer

For all session transfer modes, it is assumed that the transferee initiates the session associated with a transfer. As a pre-requisite before session initiation, it is assumed that the transferee has accepted a request for a session transfer, and/or has the necessary information to initiate a new session to handle the transferred session.

A transferee shall initiate the request for a unicast content streaming session to setup the content delivery channel and content control channel using the procedure defined in 6.3.2.1.2, with the following exceptions to Table 16.

- A new SIP header X-OITF-Replace header is included and is set to the appropriate information depending on the deployed mode. For the push mode, the information is retrieved from the incoming REFER request to the transferee as per 6.3.13.1.2.3.
- In the X-OITF request Line, the wild card part (*) representing the content instance identifier to be transferred is constructed in a different way than specified in the table. In the push mode, this field is extracted from the To header embedded in the Refer-To header in the incoming REFER request as per 6.3.13.1.2.3.

Furthermore, the IG handling in step 2 in 6.3.2.1.2 is replaced by the IG handling as outlined in 6.3.13.1.1.3.

The remaining steps in 6.3.2.1.2 apply.

6.3.13.1.1.3 IG handling of session initiation requests related to a session transfer

If the transferor OITF and the transferee OITF are behind the same IG, and given the fact that the transferee shall successfully establish the new session before the old session can be torn down, QoS resources, over the last mile, will be doubly booked while the transfer is ongoing.

Indeed, the new session initiated by the transferee to handle the transferred session may not be able to successfully complete due to resources (last mile) unavailability as a result of the old session holding on to the resources, while not being utilized, during the transfer.

In order to avoid this situation, special processing is required in the IG to release the resources associated with the transferred session while the new session is being established, if both the transferor and the transferee are behind the same IG, hence sharing the same last mile.

If for any reason the transfer failed, the IG can reclaim the released resources associated with the transferred session back and the old session can resume.

To that effect, and upon receipt by the IG for an HTTP POST for a SIP INVITE, the IG shall perform the following procedure.

- If the session is a content session associated with a session transfer and the SIP header X-OITF-Replace header is not included, then the procedure terminates.
- If the session is a content session associated with a session transfer and the SIP header X-OITF-Replace header is included and the dialog identifier points to a session whose state is not held the IG, then the transferor and the transferee are not behind the same IG and the procedure terminates.
- If the session is a content session associated with a session transfer and the SIP header X-OITF-Replace header is included and the dialog identifier points to a session whose state is held the IG, then the transferor and the transferee OITF are behind the same IG. In this case the IG shall perform the following steps:

- Step 1:** The IG shall send to the transferor OITF an HTTP 200 OK response. The response shall include a SIP re-INVITE as per Table 16 with the following exceptions.
- The SIP header information is populated appropriately given the re-INVITE is sent from the IG to the OITF.
 - The HTTP body shall be similar to SDP included by the IG in the 200 OK response to the original content session initiation request from the transferor with the port of the m-line representing the content stream being set to zero in this case.
- Step 2:** Once the OITF accepts the incoming SIP re-INVITE after the stream has been successfully paused (the transferor OITF could have already paused the stream or it would pause the stream before accepting the re-INVITE), it shall send an HTTP POST PENDING_IG request to the IG. The content of the HTTP request shall be as follows.
- HTTP request header: it includes the following:
- <list of HTTP headers> – as per IETF RFC 2616;
 - <list of SIP headers encoded as HTTP headers> – as per Table 17 (notice the direction in this case is from OITF→IG).
- HTTP request body: empty
- Step 3:** The IG shall send to the transferor OITF an HTTP 200 OK response. The response shall include a SIP ACK as per Table 14 with the following exception:
- The SIP header information is populated appropriately given the ACK is sent from the IG to the OITF.
- Step 4:** The IG shall send a SIP UPDATE (or SIP re-INVITE) to the network, to request the transferor stream to be put on hold and to reduce the requested QoS resources on the transferor OITF last mile down to zero. The SIP UPDATE (or SIP re-INVITE) shall conform to ETSI TS 124 503 in that regard. The body of the SIP UPDATE (or SIP re-INVITE) shall be identical to the SDP in the INVITE of the content session initiation with the exception described above. The IG shall wait for the response to the request.
- Step 5:** Upon receipt of a SIP 200 OK response or any other response, the procedure terminates.

6.3.13.1.2 Session transfer via push mode

6.3.13.1.2.1 OITF target discovery

An OITF that wants to locate a target OITF for session transfer purposes shall perform the procedures described in 6.4.6.1.4. Subsequently, a target OITF can be selected from the returned information.

6.3.13.1.2.2 Transferor OITF initiating a session transfer request – Push mode

To initiate a session transfer request, the transferor OITF shall follow the following procedure:

- Step 1:** As a pre-requisite, it is assumed that the user is watching a unicast content streaming service and has selected a target device (transferee OITF) for the session.
- In this step the transferor OITF shall bookmark the content as per 6.3.9.2.1. The bookmark shall be included in the body of the SIP REFER (as shown in step 2).
- Step 2:** The transferor OITF shall send an HTTP POST request for the session transfer to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.
- HTTP request header including the following:
- <list of HTTP headers> – as per IETF RFC 2616;

- <list of SIP headers encoded as HTTP headers> – as per Table 46.

HTTP request body: As per 6.3.13.2.

Step 3: The IG shall validate that the request includes all the mandatory SIP headers as per Table 46. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 4: The IG shall send a SIP REFER to the network, to setup the request and shall wait for the response. At some point in time, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the transferor OITF to report the received response from the transferee to the transfer request. The response shall include a list of SIP headers as per Table 47 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 5: Following that, the transferor OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any response reporting the outcome of the session transfer procedure.

Step 6: At some point in time, the IG shall receive an incoming SIP NOTIFY from the transferee OITF, reporting the outcome of the session transfer and which it shall forward to the transferor OITF in an HTTP 200 OK response. The HTTP response shall include the list of SIP headers as per Table 48 in addition to the normal HTTP headers. The body of the HTTP response shall include the SDP body received in the NOTIFY.

Step 7: The transferor OITF shall return the SIP 200 OK response, acknowledging the SIP NOTIFY, to the IG, in an HTTP POST PENDING_IG request. The content of the HTTP request shall be as follows.

HTTP request header including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 49.

HTTP request body: Empty

Table 46 lists the details for HTTP extension headers for an outgoing SIP REFER from the transferor for initiating up a session transfer request (OITF→IG) and incoming SIP REFER request to the transferee (IG→OITF).

Table 46 –HTTP extension headers for SIP REFER

X-OITF HTTP header	Source of coding information
X-OITF-request-Line NOTE The request URI shall be set to the transferee (target device OITF) contact information received during the device discovery process.	IETF RFC 3261 REFER <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Refer-To shall be set to the remote target URI included in the contact header field returned in the SIP 200 OK associated with initial session setup with the transferor and extended with the following URI headers fields: Replaces header field shall include the SIP dialog identifier for the original unicast content streaming session as per IETF RFC 3891 Require header field populated with the option tag value "replaces" To header field shall contain the original content identifier copied from the request URI of the original SIP INVITE request initiated from the transferor. OPTIONALLY an Accept-Contact header field with the g.3gpp.icsi-ref media feature tag set to the IPTV Communication service identifier "urn%3Aurn-7%3A3gpp-service.ims.icsi.iptv" P-preferred-service set to IPTV Communication service Identifier urn:urn-7:3gpp-service.ims.icsi.iptv body header. Contains the SDP body to be included in the SIP request initiated from the transferor. OITF. The SDP body shall contain the same number of media lines as the SDP used in the original session from the transferor OITF. Each media line shall indicate the same media type as its corresponding media component in the SDP used in the original session by the transferor OITF. The media line for the media to be transferred shall include a port number with non zero value. EXAMPLE: Refer-To: <sip:remoteuser@home2.net;gr=urn:uuid:f81d4fae-7dec-11d0-a765-333333333333?Replaces=AB03a0s09a2sdfgIkJ490333%3Bremote_tag=Afgsdfg45%3Blocal-tag=U188gg&Require=replace&P-Preferred-service=urn:urn-7:3gpp-service.ims.icsi.iptv&Accept-Contact=%3b+g.3gpp.icsi-ref%3d%22urn%253Aurn-7%253gpp-service.ims.icsi.iptv%22	IETF RFC 3261 ETSI TS 124 237 IETF RFC 3891
X-OITF-Contact	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.session-transfer+xml" corresponding to 6.3.13.2.	
X-OITF-Content-Length	IETF RFC 3261

Table 47 lists the details of HTTP extension headers for the response to an outgoing SIP REFER from the transferor for setting up a session transfer request (IG→OITF) and the response sent to an incoming SIP REFER to the transferee (OITF→IG).

**Table 47 –HTTP extension headers
SIP REFER response**

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

6.3.13.1.2.3 Transferee OITF receiving an incoming session transfer request – push mode

The procedure at an OITF selected to be the target device (transferee OITF) in a push mode is as follows.

- Step 1:** It is assumed that the OITF has an HTTP PENDING_IG request. At some point in time, when a REFER request targeted for the transferee OITF is received by the IG, the IG shall return a HTTP 200 OK response to the OITF. The response shall include the list of SIP headers as per Table 46, in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the XML structure as per 6.3.13.2.
- Step 2:** The OITF shall examine the incoming REFER request. In particular, the OITF shall extract the body header to use it to later construct its own SDP for the session transfer (see 6.3.13.1.1.2). If the OITF cannot successfully validate the extracted SDP, it shall reject the incoming request. If the OITF successfully validates the extracted SDP, it should accept the incoming request.
- Step 3:** Once the OITF accepts the incoming SIP REFER, it shall send an HTTP POST PENDING_IG request to the IG. The content of the HTTP request shall be as follows:
- HTTP request header: It includes the following:
- <list of HTTP headers> – as per IETF RFC 2616,
 - <list of SIP headers encoded as HTTP headers> – as per Table 47 with the exception that the response in this case is a SIP 202 OK
- HTTP request body: empty
- Step 4:** The OITF shall extract the following information from the incoming REFER request:
- The Content URI extracted from the To header included in the Refer-To header.
 - The body header.
 - The Dialog ID to be replaced extracted from the Replace header in the Refer-To header.
 - The content bookmark from the HTTP body, if present, and if understood by the OITF.
- Step 5:** The transferee OITF shall then construct an SDP that it can use to initiate a new session to handle the transfer. The OITF may follow 6.3.2.1.1, if need be, towards the construction of the SDP.
- Step 6:** The transferee OITF shall then invoke the procedure defined in 6.3.13.1.1.2.
- Step 7:** Once the session setup is successfully completed, the OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 48.

HTTP request body: SIP/2.0 200 OK (or the outcome of the session initiation request)

Step 8: The IG shall validate that the request includes all the mandatory SIP headers as per Table 52. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 9: The IG shall send a SIP NOTIFY to the network, to report the outcome of the session transfer and shall wait for the response. The IG shall return a HTTP 200 OK response (or other appropriate responses) to the transferee OITF to report the received response from the transferor OITF. The response shall include a list of SIP headers as per Table 49 in addition to the normal HTTP headers as per IETF RFC 2616.

Following that, all unicast content streaming procedures apply to the session.

Table 48 lists the supported HTTP extension headers in the NOTIFY request sent from the transferee OITF (OITF→IG) to the transferor (IG→OITF).

Table 48 –HTTP extension headers in the NOTIFY request

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP REFER	IETF RFC 3261 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3515
X-OITF-Call-ID	IETF RFC 3261
X-OITF-Subscription-State	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "message/sipfrag"	IETF RFC 3515, IETF RFC 3420
X-OITF-Length	IETF RFC 3261

Table 49 lists the supported HTTP extension headers in the response to a NOTIFY request received from the IG (IG→OITF) transferor to the transferee.

Table 49 –HTTP extension headers in the NOTIFY response

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

6.3.13.2 XML schema for session transfer information included in a session transfer request from the transferor to transferee

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:transfer:2011"
  xmlns:tns="urn:oipf:iptv:transfer:2011"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:bmk="urn:oipf:iptv:bookmark:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd" />
  <xs:import namespace="urn:oipf:iptv:bookmark:2011"
    schemaLocation="iptv-bookmark.xsd"/>
  <xs:element name="sessiontransfer">
    <xs:annotation>
      <xs:documentation>This describes information elements needed to support
        session transfer</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="session-bookmark" type="bmk:BookmarkType"/>
      <xs:element name="transferee" type="ct:UserIdType" />
      <xs:documentation> this element is populated with the same
        information
        included in the request URI of the REFER request
      </xs:documentation>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
        maxOccurs="unbounded" element name="any" type="any"/>
    </xs:sequence>
  </xs:element>
</xs:schema>
```

6.4 Protocol for service access and control functions

6.4.1 Service provider discovery

6.4.1.1 Protocol over HNI-IGI – HTTP option

6.4.1.1.1 Retrieval of service provider discovery information

The procedures in this subclause shall only be performed in the context of the default user. When the OITF supports native HNI-IGI, it shall follow the following procedure to retrieve service provider discovery information:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 50.

HTTP request body: empty or optionally, the OITF may include a body associated with the appid “urn:oipf:application:iptv-SP-discovery”. The optional message body sent to the service provider discovery FE shall include the capabilities of the OITF. The Content-Type of the message body shall be set to “application/vnd.oipf.ueprofile+xml”, which refers to the MIME type of the schema defined in Clause B.2. See Table 50 for X-OITF-Content-Type header.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers required for the outgoing message as per Table 50. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

- Step 3:** If the IG has the requested information, it shall respond immediately with HTTP 200 OK. If not, the IG shall send a SIP SUBSCRIBE to the network, to subscribe to the “ua-profile” event, and shall wait for the response to the subscription request. The IG shall then return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the subscription request. The response includes a list of SIP headers as per Table 51 in addition to the normal HTTP headers as per IETF RFC 2616.
- Step 4:** The OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any incoming messages.
- Step 5:** When a SIP NOTIFY is received by the IG for a “ua-profile” event, the IG shall return an HTTP 200 OK response to the OITF. The response includes a list of SIP headers as per Table 52 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall be the SIP body received in the incoming NOTIFY message. The content of the HTTP response shall be as follows:
- HTTP response header: Including the following:
- <list of HTTP headers> – as per IETF RFC 2616;
 - <list of SIP headers encoded as HTTP headers> – as per Table 52.
- HTTP response body: body of the incoming NOTIFY.

The OITF shall parse the XML document in the body to ensure that it complies with the schema defined in 3.2.1 of IEC 62766-3:2016.

When parsing the list of parameters, the OITF shall take the following action.

- If the service provider discovery information for a service provider is already present in the OITF (i.e., for which the OITF already has an entry), and
 - if the “@Version” attribute does not have the same value as that received in the NOTIFY message, then the OITF shall perform the following actions:
 - The OITF shall update its parameters with the new values sent by the service provider discovery FE. Also if the Segment@ID or Segment@Version has changed, the OITF shall update the service discovery information with that received from the service discovery FE.
 - If the “@Version” attribute has the same value as that received in the NOTIFY message, the OITF shall not update the stored service provider discovery information.
- If the service provider discovery information for a service provider is not known to the OITF (i.e., the OITF does not have an entry for the service provider discovery information):
 - The OITF shall create a new entry for the new service provider with all the parameters received in the NOTIFY message.

The IPTV service provider discovery information delivered via this protocol shall conform to ETSI TS 102 034, Part 3, 5.2.5, with the extended element defined in the Metadata Specification Part 3.

- Step 6:** Once the OITF accepts the HTTP message containing the incoming SIP NOTIFY, it shall send an HTTP HNI-IGI PENDING_IG request to the IG. The content of the HTTP request shall be as follows:
- HTTP request header: Including the following:
- <list of HTTP headers> – as per IETF RFC 2616;
 - <list of SIP headers encoded as HTTP headers> – as per Table 53.
- HTTP request body: empty
- Step 7:** The IG shall send the SIP 200 OK response to the network and then shall return to Step 5 to handle any subsequent NOTIFY received from the network.

6.4.1.1.2 Procedure for cancellation of the subscription

The procedure for de-registering the IPTV default user shall be preceded with a cancellation of subscription.

The procedure is the same as the procedure for initiating a subscription to the “ua-profile”, except that the X-OITF-Expires header in Table 50 shall be set to 0.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

Table 50 – Supported HTTP extension headers in HNI-IGI SUBSCRIBE request for SP discovery

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI shall be set to the well known PSI. The PSI shall be composed of the domain name extracted from the public user identity with a user part set to "OIPF_IPTV_SPD". (e.g., OIPF_IPTV_SPD@<domain_name>)	IETF RFC 3261 SUBSCRIBE <request URI> SIP/2.0
X-OITF-From Note that the From user shall be set to the IMPU of the default user.	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Event Extend the existing "ua-profile" event package for SIP SUBSCRIBE request The Event header shall be set to the "ua-profile" event package. The Event parameters shall be set as follows: The "profile-type" parameter shall be set to "application". The "appids" parameter shall be set to "urn:oipf:application:iptv-SP-discovery".	IETF RFC 3265 and as per ETSI TS 183 063, 5.1.2.2.1
X-OITF-Contact Remarks: a) URI parameter shall be included, and shall match the value that is sent in the Contact header in the registration request. b) Expires parameter should be included. c) Priority parameter should be included. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires Note that if absent a default value according to IETF RFC 3261 shall be assumed by the IG. To cancel the subscription, the X-OITF-Expires shall be set to 0	IETF RFC 3261
X-OITF-Accept Set to "application/vnd.oipf.spdiscovery+xml"	IETF RFC 3261
X-OITF-Content-Type OPTIONALLY included when signalling OITF capabilities according schema defined in Clause B.2. shall be set to "application/vnd.oipf.ueprofile+xml"	IETF RFC 3261

Table 51 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request for SP discovery

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 52 – Supported HTTP extension headers in the NOTIFY request to the SUBSCRIBE to SP discovery

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP SUBSCRIBE	IETF RFC 3261, IETF RFC 3265 and IETF RFC 6080 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3265 and as per ETSI TS 183 063 Subclause 5.2.2.2
X-OITF-Call-ID	IETF RFC 3261
X-OITF-Subscription-State	IETF RFC 3265 and IETF RFC 3856
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/vnd.oipf.spdiscovery+xml"	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

Table 53 – Supported HTTP extension headers in the response to a NOTIFY request to the SUBSCRIBE to SP discovery

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

NOTE Cancellation of subscription is not required if the X-OITF-Expires header was set to 0 in the initial SUBSCRIBE request.

6.4.1.1.3 Refreshing the subscription

The procedure for refreshing a subscription is the same as the procedure for initiating a subscription.

The application initiating the subscription procedure shall refresh the subscription based on the refresh subscription timer information received in the response to the subscription. Refreshing a subscription should be performed before the expiry of the refresh timer. A subscription that is not refreshed will be terminated.

The IG shall consider a subscription terminated if it is not refreshed.

6.4.1.2 Protocol over UNIS-19 and non-native HNI-IGI

The OITF retrieves the service provider discovery entry point and uses the entry point to retrieve a list of IPTV service providers using HTTP for that purpose. The IPTV service providers list shall be delivered as SD&S records or DAE applications.

When an IPTV service provider discovery entry point is selected, service provider discovery information shall be delivered as service discovery and Selection (SD&S) records or as DAE applications. This information is provided by the service platform provider.

When SD&S records are used, the HTTP protocol conforming to ETSI TS 102 034, 5.4.2 shall be used for the transport of IPTV service provider discovery information. The data delivered shall conform to ETSI TS 102 034, 5.2.5, with the extension defined in IEC 62766-3.

When DAE applications are used, the HTTP protocol and data formats shall conform to 6.3.1.7 of IEC 62766-5-1:2017.

6.4.2 Service discovery

6.4.2.1 Protocol over UNIS-6

The protocol on UNIS-6 shall be HTTP as defined in IEC 62766-5-1 for DAE application based service discovery. This protocol is used for the unicast transport of HTML ECMAScript documents between the OITF DAE function and the IPTV application functional entity.

6.4.2.2 Protocol over UNIS-15

The protocol used on UNIS-15 for the transport IPTV service discovery information shall be HTTP conforming to ETSI TS 102 034, 5.4.2.

The IPTV service discovery information delivered via this protocol shall conform to ETSI TS 102 034, 5.2.6 with the extension defined in IEC 62766-3.

6.4.3 Service access

6.4.3.1 Protocol over UNIS-6

UNIS-6 may be used for the unicast transport of HTML ECMAScript documents between the OITF DAE function and the IPTV application functional entity for DAE application based service access.

See IEC 62766-5-1 for the details of the document format delivered via this protocol.

6.4.3.2 Protocol over UNIS-7

The use of the HTTP protocol on this reference point shall comply with 4.1.2.2.2 (container based delivery) or 4.2 (query mechanism) of the DVB-IP Broadband Content Guide specification (ETSI TS 102 539).

The Content Guide metadata delivered via this protocol shall conform to ETSI TS 102 539 (ETSI TS 102 539) with the extension defined in IEC 62766-3.

The OITF may request user specific information from the Metadata control FE based on the IPTV Subscription profile. (See 6.4.4).

6.4.4 Subscription profile management and usage

6.4.4.1 Protocols on UNIP-1 for XCAP-based profile management

6.4.4.1.1 General

The OITF shall be able to obtain a user's IPTV Subscription profile. The format of the IPTV Subscription profile shall conform to Clause B.1. The IPTV Subscription profile may be used for filtering the Broadband Content Guide metadata, i.e. for the provision of a personalised content guide.

The IPTV service profile functional entity shall expose XCAP Server behaviour (HTTP Server 1.1, XML parser, and data repository) as defined in IETF RFC 4825.

UNIP-1 shall comply with XCAP as defined in IETF RFC 4825.

6.4.4.1.2 XCAP application usage for IPTV service

6.4.4.1.2.1 Profile management

The XML Configuration Access Protocol (XCAP) defined in IETF RFC 4825 is used for manipulating data stored in the IPTV service profile functional entity. XCAP allows a client to read, write and modify application configuration data, stored in XML format, on a server. XCAP maps XML document sub-trees and element attributes to HTTP URIs, so that these components can be directly accessed by HTTP. XCAP uses the HTTP methods PUT, GET, and DELETE to operate on documents stored in the service profile functional entity.

The data stored in the IPTV service profile functional entity relates to the operation of the IPTV service. This specification defines a new application usage to allow a client to manipulate data related to IPTV services.

XCAP requires the definition of XML documents that are compliant with the XML schema and constraints defined for a particular XCAP application usage. The application usage defines the XML schema for the data used by the application, along with other key pieces of information.

Central to XCAP is the construction of the HTTP URI that points to a particular document or certain components of it. A component in an XML document can be an XML element, attribute, or the value of it.

6.4.4.1.2.2 XCAP application usage

XCAP requires application usages to fulfil a number of steps in the definition of such application usage. The remainder of this subclause specifies the required definitions of the IPTV services XCAP application usage.

Application Unique ID (AUID): Each XCAP application usage is associated with a unique name called the application Unique ID (AUID). The AUID defined by this application usage falls into the vendor-proprietary namespace of XCAP AUID, where Open IPTV Forum is considered a vendor.

The proposed AUID to be allocated to the Open IPTV Forum IPTV services application usage shall be:

org.openiptvforum.iptv

XML schema: Implementations in compliance with this specification shall implement the XML schema defined in Annex B.

Default namespace: XCAP requires application usages to declare the default namespace. The default namespace of the IPTV services XCAP application usage shall be:

urn:oipf:params:xml:ns:iptv

MIME Type: The MIME type of IPTV service XML document shall be:

application/vnd.oipf.userprofile+xml

Validation constraints: This specification does not specify any additional constraints beyond those defined by XCAP.

Data semantics: The XML schema does not accept URIs that could be expressed as a relative URI reference causing a resolution problem. However, each of the supplementary services should consider if relative URIs are allowed in the subdocument tree, and in that case, they should indicate how to resolve relative URI references. In the absence of further indications, relative URI references should be resolved using the document URI as the base of the relative URI reference.

Naming conventions: By default, IPTV service profile XML documents are stored in the IPTV service profile functional entity. In order to facilitate the manipulation of an IPTV service profile XML document, the default XML file name shall be:

iptvprofile.xml

Resource interdependencies: This specification does not specify additional resource interdependency beyond those specified in the XML schema.

Authorization policies: The authorization policy for access and manipulation of an IPTV service profile document shall be defined by the service provider.

6.4.4.2 Protocols over UNIS-6 for DAE-based profile management

UNIS-6 may be used for the unicast transport of HTML ECMAScript documents between the OITF DAE function and the IPTV application FE for DAE-based subscription profile management. In this case, the IPTV application FE acts as a front-end to the IPTV service profile FE. When the HTTP request for profile management is received from OITF, the IPTV application FE manipulates the IPTV service profile FE.

6.4.4.3 Protocols over HNI-IGI – HTTP option

6.4.4.3.1 Subscription to notification of changes in the IPTV service profile

The procedure for subscription to notification of changes in the IPTV service profile shall be invoked from either a DAE application or an embedded application in the OITF. The procedures shall be as follows:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: includes the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 54.

HTTP request body: the body contains the list of the requested URIs associated with the XCAP resources for which the subscription is issued. The MIME Type of the document inserted in the body will be signalled by the Content-Type header, set to "application/vnd.oipf.userprofile+xml".

Step 2: The IG shall validate that the request includes all the mandatory SIP headers needed for the outgoing subscription message, as per Table 54. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall send a SIP SUBSCRIBE to the network, to subscribe to the "xcap-diff" event package, and shall wait for the response to the subscription request. The IG shall return a HTTP 200 OK response to the OITF to report the response to the subscription request. The response shall include a list of SIP headers as per Table 55 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 4: The OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any incoming messages.

Step 5: When a SIP NOTIFY is received by the IG, the IG shall return a HTTP 200 OK response to the OITF that includes the information carried in the incoming NOTIFY. The response shall include a list of SIP headers as per Table 56 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the "xcap-diff+xml" document carried in the NOTIFY body. This document contains the changes in the XCAP document(s) identified in the subscription request in Step 1(b).

Step 6: When the OITF accepts the incoming SIP NOTIFY, it shall send an HTTP POST PENDING_IG request to the IG to acknowledge the receipt of notification. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 57.

HTTP request body: empty

Step 7: The IG shall send the SIP 200 OK response to the network and then shall return to Step 5 to handle any subsequent NOTIFY messages that may be received from the network.

Table 54 – Supported HTTP extension headers in HNI-IGI SUBSCRIBE request for receiving notification of changes in the IPTV service profile

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI shall be set to the well-known PSI of the IPTV service profile FE: The PSI shall be "OIPF_IPTV_serviceprofile@<domainname>" where <domainname> shall be the IPTV service provider domain name obtained through service provider discovery.	IETF RFC 3261, IETF RFC 3265 and IETF RFC 5875 SUBSCRIBE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Event The Event header shall be set to the "xcap-diff" event package.	IETF RFC 3265 and as per ETSI TS 183 063, 5.1.5.1
X-OITF-Accept The Accept header shall include the value "application/xcap-diff+xml". This header indicates the body formats allowed in subsequent NOTIFY requests	IETF RFC 3265 and as per ETSI TS 183 063, 5.1.5.1
X-OITF-Content-type shall be set to "application/vnd.oipf.userprofile+xml" as the MIME Type of IPTV Subscription profile schema.	IETF RFC 3265
X-OITF-Contact Remarks: a) The URI parameter shall be included and shall match what is sent in the Contact header included in the registration request. b) The Expires parameter should be included.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires Note that if absent a default value shall be assumed by the IG.	IETF RFC 3261

Table 55 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 56 – Supported HTTP extension headers in the NOTIFY request containing changes in the IPTV service profile

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP SUBSCRIBE	IETF RFC 3261 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3265 and as per ETSI TS 183 063, 5.1.5.2
X-OITF-Call-ID	IETF RFC 3261
X-OITF-Subscription-State	IETF RFC 3265
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type shall be set to "application/xcap-diff+xml".	IETF RFC 3265 and as per ETSI TS 183 063, 5.1.5.2

Table 57 – Supported HTTP extension headers in the response to a NOTIFY request

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

6.4.4.3.2 Refreshing the subscription

It is the responsibility of the application initiating the subscription procedure to refresh the subscription according to the "refresh subscription timer" parameter received in the response to the subscription request. Refreshing the subscription should be performed before the expiry of the refresh timer. A subscription that is not refreshed shall be terminated after the expiration of the timer.

The IG shall consider a subscription terminated if it is not refreshed.

6.4.4.3.3 Procedure for cancellation of a subscription

This procedure may be invoked at any time.

The procedure for de-registering the IPTV end user shall be preceded by the cancellation of any subscription for notification of changes in the user's IPTV service profile.

The procedure for cancellation of the subscription is the same as the procedure for initiating a subscription to the ua-profile event package, except that the X-OITF-Expires header in Table 54 shall be set to 0.

6.4.5 Remote management

6.4.5.1 General procedures on UNI-RMS

6.4.5.1.1 General

The remote management functions required for managed devices are specified in the general framework document TR-069 by the Broadband Forum. The framework document is associated with a number of technical reports that define the CWMP data models that are specific for each device function.

6.4.5.1.2 UNI-RMS for IG, AG and WAN Gateway

In addition to TR-069, the following specifications shall apply:

- TR-098 that defines the data model for the “internet gateway device” shall apply to the WAN Gateway FE (see RMS3 functional block of the OIPF functional Architecture, see IEC 62766-1:2016, Annex B).
- TR-106 that defines the data model for the generic CWMP-managed device shall apply to the IG, AG and WAN-Gateway FEs.
- TR-104 that defines the data model for the “SIP end-point” shall apply to the IG (see RMS2 functional block of the OIPF functional Architecture, see IEC 62766-1:2016, Annex B).

6.4.5.1.3 UNI-RMS for OITF

6.4.5.1.3.1 General

Although the remote management functions are specified in the general framework document TR-069 by the Broadband Forum, the protocol to remotely manage OITF retail devices is intended to support limited functions mainly for Performance Monitoring and Diagnostics. Consequently, an OITF device doesn't fulfil all the requirements that are requested in TR-069. The limitations outlined in the following subclauses shall apply.

6.4.5.1.3.2 OITF RPC methods support requirements

An OITF shall implement the following RPC methods as listed in Table 58.

Table 58 – RPC methods

Method name	OITF requirement	ACS requirement
CPE methods	responding	calling
GetRPCMethods	required	required
SetParameterValues	required	required
GetParameterValues	required	required
SetParameterAttributes	required	optional
GetParameterAttributes	required	optional
ACS methods	calling	responding
Inform	required	required

As an OITF device is not required to support all the RPC requirements as defined in TR-069, the ACS shall implement the GetRPCMethods to discover the limited set of methods supported by the OITF.

The OITF RPC Methods shall respect the calling arguments and type as defined in TR-069, with the following definition of the DeviceIdStruct that is used for the DeviceId argument of the Inform method:

The three parameters ManufacturerOUI, ProductClass and Serial Number have slightly different semantic meanings in the context of OIPF and are obtained from the deviceId identifier (refer to 7.2.4.2.1).

- ManufacturerOUI = HEX(first 3 bytes of SHA-1(X))
- ProductClass = "OIPF"
- SerialNumber = HEX (remaining bytes, from 4th on, of SHA-1(X))
where, X = (MAC address as bytes) + (domain name in ASCII characters).

Table 59 summarises the characteristics of the RPC method parameters.

Table 59 – RPC method parameters

Name	Type	Description
Manufacturer	String(64)	Manufacturer of the device
OUI	String(6)	In the context of OIPF, this parameter is the hexadecimal value of the first 3 bytes of SHA-1(X)
ProductClass	String(64)	In the context of OIPF, this parameter is always "OIPF"
SerialNumber	String(64)	In the context of OIPF, this parameter is the hexadecimal value of the remaining bytes (from 4th on) of SHA-1(X)

6.4.5.1.3.3 OITF data model

In the framework of the Open IPTV Forum, a specific data model for the remote management of a retail OITF device has been defined. The data model has been obtained from TR-135 and TR-106 with a selection of a reduced set of parameters using the same semantics (with a few exceptions) and the same types. The OITF data model is fully described in Annex K.

6.4.5.1.4 Configuration of the IG via configuration file

6.4.5.1.4.1 General

CPE WAN Management protocol based on Broadband Forum TR-069 shall be used to configure the IPTV application in the IG. An IPTV configuration file shall be used to populate the IG with the list of users with their IMPU, Alias and passwords and also configure whether user authentication is to be performed by the IG.

If GBA authentication or HTTP digest authentication is supported by the IG, the IG shall be configured with the following information:

- whether it has to provide an intended identity or not in the GBA and HTTP digest authentication procedures as described in IEC 62766-7;
- the realm, username and password for regular HTTP digest authentication.

The file is downloaded to the IG during the IG power up procedure.

The configuration data shall be defined in XML and shall include the XML schema to be enforced against the configuration data.

6.4.5.1.4.2 Call flow

There are 2 cases to be considered:

In the first case, the remote server requests the IG to download the configuration file at power up of the IG. This requires the IG to contact the remote server. The download request is subsequently used by the server to request the IG to download the configuration file. Alternatively, if the server is configured (by some means) with the address of the IG, it can request the IG to contact it using the Connection request Notification mechanism, if the remote server supports this mechanism.

In the second case the process is initiated by the IG if it detects a corrupted file or if for some reason it lost the file due to a reboot or an internal error.

Figure 4 is a call flow depicting the configuration procedure.

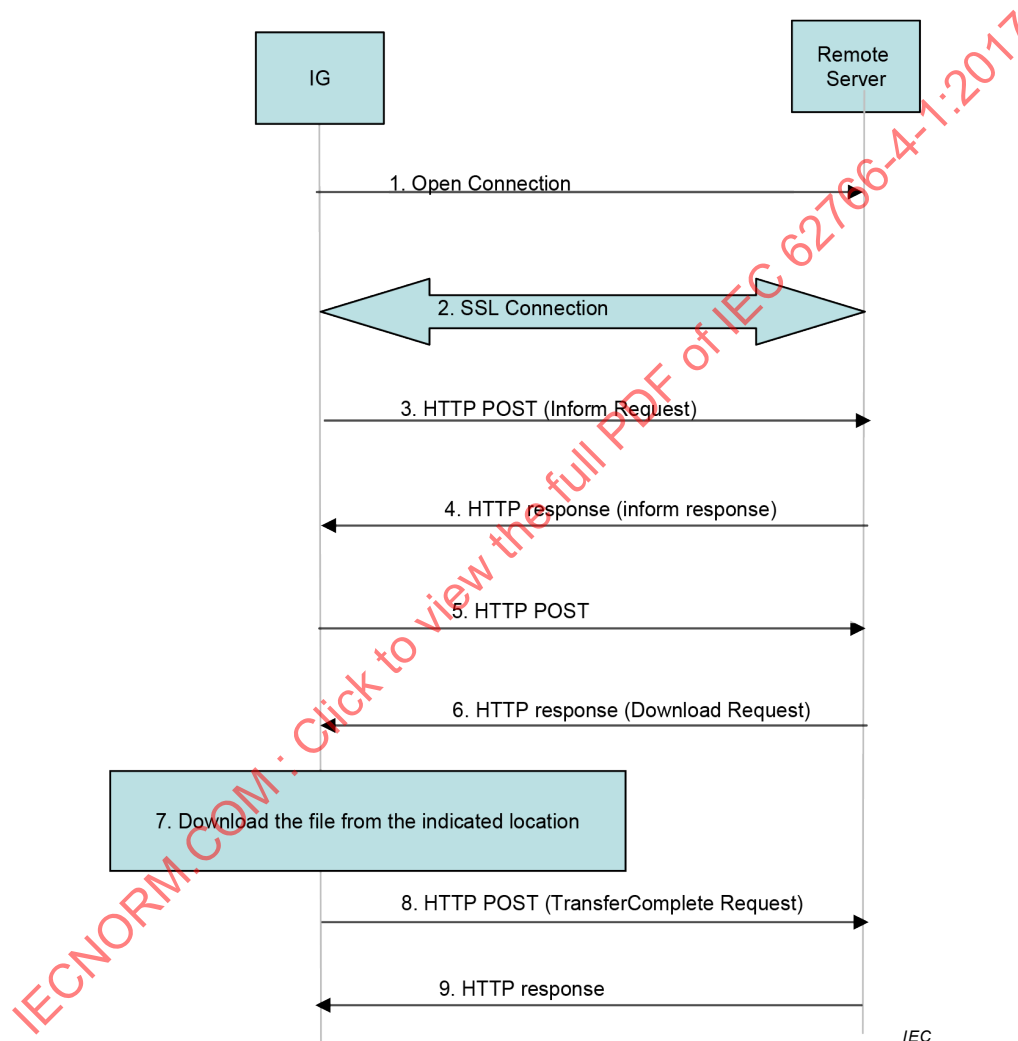


Figure 4 – Sequence for the configuration of an IG

The following is a brief description of the flow:

Steps 1-4: Normal steps as per TR-069.

Step 5: The IG sends an HTTP POST request with no HTTP entity body to the remote server.

Step 6: The server returns an HTTP response that includes a Download request in the HTTP entity body. The arguments are set as follows:

CommandKey: Mandatory – set by remote server.

FileType: Mandatory – set to 3: Vendor Configuration File. The vendor in this case is Open IPTV Forum.

URI: Mandatory – set by remote server

Username: OPTIONAL – If used, shall be configured in the IG and remote server

Password: OPTIONAL – If used, shall be configured in the IG and remote server

TargetFileName: Mandatory – IPTV-ConfigurationParameters

DelaySeconds: Mandatory – set to no delay

Successful URI: Not provided

Failure URI: Not provided

Step 7: Following that, the IG proceeds to download the configuration file.

Step 8-9: Once the download is complete, the IG sends a TransferComplete request to the remote server. The arguments in the request are set as follows:

CommandKey: Mandatory – set to the value received in the download request.

FaultStruct: Mandatory in case of failure according to TR-069

StartTime: Mandatory – set according to TR-069

FinishTime: Mandatory – set according to TR-069

NOTE The above sequence is an example and there are other valid sequences that can achieve the same result.

6.4.5.1.4.3 XML schema for the IPTV configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:config:ig:2009"
  xmlns:tns="urn:oipf:config:ig:2009"
  xmlns:enum="urn:ietf:params:xml:ns:enum-token-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema">

  <!--schema filename is config-ig.xsd -->
  <xs:annotation>
    <xs:documentation xml:lang="en">
      This schema is copyrighted by the Open IPTV Forum ("OIPF") and
      distributed in conjunction
      with Release 1 of the IPTV Solution Specification.

      Disclaimer
      The Open IPTV Forum members accept no liability whatsoever for any
      use of this document.
      This specification provides multiple options for some features.
      The Open IPTV Forum Profiling
      specification will complement the Release 1 specifications by
      defining the Open IPTV Forum
      implementation and deployment profiles. Any implementation based
      on Open IPTV Forum
      specifications that does not follow the Profiling specifications
      cannot claim Open IPTV Forum
      compliance.

      Copyright Notification
      No part may be reproduced except as authorized by written
      permission.
      Any form of reproduction and/or distribution of these works is
      prohibited.
      Copyright 2009 © Members of the Open IPTV Forum
      All rights reserved.
    </xs:documentation>
  </xs:annotation>
```

```

<xs:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="xml.xsd" />
<xs:import namespace="urn:ietf:params:xml:ns:enum-token-1.0"
  schemaLocation="imports/enum-token-1.0.xsd" />

<xs:element name="IGconfiguration" type="tns:IGconfigurationType" />
<xs:complexType name="IGconfigurationType">
  <xs:sequence>
    <xs:element name="AuthenticationSet"
      type="tns:AuthenticationSetType" maxOccurs="unbounded" />
    <xs:element name="GatewayAuthentication" type="xs:boolean"
      minOccurs="0" />
    <xs:any namespace="##other" processContents="skip"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthenticationSetType">
  <xs:sequence>
    <xs:element name="Identifier" type="tns:IMSPublicIdType" />
    <xs:element name="Password" type="xs:string" />
    <xs:element name="Alias" type="xs:string" />
    <xs:sequence minOccurs="0">
      <xs:element name="IMPI" type="xs:string" />
      <xs:element name="SIPDigestPassword" type="xs:string" />
    </xs:sequence>
  </xs:sequence>
</xs:complexType>

<!-- ===== Definition for IMSPublicIdType =====>
<xs:complexType name="IMSPublicIdType">
  <xs:choice>
    <xs:element name="e164Number" type="enum:e164numberType" />
    <xs:element name="SIPURI" type="tns:SIPURITYPE" />
  </xs:choice>
</xs:complexType>

<xs:simpleType name="SIPURITYPE">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      SIP URI pattern is defined based on the SIP URI
      description provided in IETF RFC 3261 (Subclause 2)
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern
      value="[sS][iI][pP][sS]?:(//([^/?#]*)?)?([?#](\?([^#]*)?)(#.*)?)?"
    />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

The schema establishes a binding between an IMS Public Identity (IMPU), a user alias and a password. If SIP digest authentication is used for user authentication, the IMPI and SIPDigestPassword shall be included. Otherwise, it is assumed that user authentication is based upon IMS AKA.

The schema also supports a mechanism to instruct the IG if user authentication is mandatory in the consumer network.

The schema is extensible.

An example of a configuration file that conforms to the above schema is as follows:

```
<IGconfiguration>
  <AuthenticationSet>

<Identifier><SIPURI>sip://operator.example.com/MickJ</SIPURI></Identif
ier>
  <Password>RollingStones</Password>
  <Alias>Mick Jagger</Alias>
  <IMPI>household123@operator.com</IMPI>
  <SIPDigestPassword>CCXDFGGH</SIPDigestPassword>
</AuthenticationSet>
<AuthenticationSet>

<Identifier><SIPURI>sip://operator.example.com/BruceS</SIPURI></Identif
fier>
  <Password>TheBoss</Password>
  <Alias>BruceSpringstein</Alias>
  <IMPI>household123@operator.com</IMPI>
  <SIPDigestPassword>CCXDFGGH</SIPDigestPassword>
</AuthenticationSet>
  <GatewayAuthentication>true</GatewayAuthentication>
</IGconfiguration>
```

6.4.5.2 Remote management using DAE APIs

See the DAE Specification, IEC 62766-5-1:2017, 7.11.

6.4.6 User registration and network authentication

6.4.6.1 Procedure for user registration and authentication in network relying on IMS on the HNI-IGI interface – HTTP option

6.4.6.1.1 User registration

This procedure shall be invoked in following cases:

- when the OITF is turned on or restarted if the OITF supports the native HNI-IGI function;
- when an IPTV end user explicitly logs on at an OITF using an alias or IMPU or a default IMPU.

The IG shall extract the deviceId from the sip instance feature tag.

If the deviceId and the IMPU match another deviceId and IMPU whose state is held in the IG, the IG shall conclude that the OITF has undergone a restart and shall proceed to immediately clear all SIP sessions belonging to the OITF. Following that, the IG shall de-register all users registered from that OITF.

If GRUU is not requested, the IG shall not perform IMS registration when the IMPU is already registered. However, the IG shall maintain a binding between the Alias/IMPU, the OITF device from which the registration is received (extracted from the sip instance feature tag) and the new contact information including the sip instance feature tag, which provides an easy way to guarantee uniqueness within the Address of Record (AOR).

Following the successful registration of the IMPU as per the procedure below, the IG shall maintain a binding between the Alias/IMPU, the OITF device (extracted from the sip instance feature tag described above) from which the registration is received, and the new contact

information including the sip instance feature tag, which provides an easy way to guarantee uniqueness within the Address of Record (AOR).

If the identity being registered is the default identity, and if the default identity is not bound to any OITF in the consumer network, then the IG shall deregister its contact address for the default identity at the end of this procedure.

If the identity being registered is not the default identity and if the default identity is not bound to any OITF in the consumer network, then the IG shall deregister the default identity from the IG point of view at the end of this procedure.

Step 1: The OITF shall send an HTTP POST request to the IG on the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <List of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 60

HTTP request body: empty

Step 2: The IG shall validate that the request includes all the mandatory SIP headers needed for the outgoing registration message, as per Table 60. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for the rejection.

Step 3: Once the IG completes the IMS registration process, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF. The response shall include a list of SIP headers as per Table 61 in addition to the normal HTTP headers as per IETF RFC 2616.

If the OITF does not support native HNI-IGI, user registration shall be done through a DAE application.

6.4.6.1.2 User de-registration

This procedure is invoked in the following cases.

- The OITF is turned off and the OITF supports native HNI-IGI.
- An IPTV end user, who has registered with his own IMPU or a default IMPU, deregisters from an OITF.

Prior to de-registering the IMPU, the IG shall clear all SIP sessions in which the IMPU is engaged on the specific OITF device from which the de-registration occurred and shall subsequently remove all bindings between the IMPU and all SIP sessions on the impacted OIPF. Following successful deregistration of the IMPU, the IG shall remove the binding between the IMPU and the OITF device from which the de-registration has occurred.

If GRUU is not supported for this registration, the IG shall not perform IMS deregistration when an IMPU is already registered on multiple OITFs, but the IG shall remove the binding between the IMPU and the OITF from which the user has deregistered (extracted from the sip instance feature tag) including the contact information (including the sip instance feature tag).

If GRUU is not supported for this registration, the IG shall perform the IMS deregistration procedure if the IMPU was bound to a single OITF.

Following a successful de-registration, the IG shall remove the binding between the Alias/IMPU, the OITF device from which the registration is received (extracted from the sip instance feature tag).

Following the successful de-registration of the IMPU, if there are no more OITFs still turned on in the consumer network, the IG shall re-register the default identity from the IG point of view.

Step 1: The OITF shall send to the IG an HTTP POST request containing an X-OITF-request-Line header on the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.

HTTP request header including the following:

- <list of HTTP headers> – as per IETF RFC 2616,
- <list of SIP headers encoded as HTTP headers> – as per Table 60.

HTTP request body: empty

Step 2: The IG shall validate that the request includes all the mandatory SIP headers needed for the outgoing de-registration message as per Table 60. The IG shall reject any request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for the rejection.

Step 3: Once the IG completes the IMS de-registration process, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF. The response shall include a list of SIP headers as per Table 61 in addition to the normal HTTP headers as per IETF RFC 2616.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

**Table 60 –Mandatory HTTP extension headers
for user registration/de-registration (OITF→IG)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI is that of the P-CSCF, and is fetched by the OITF as per 7.1.1 of ETSI TS 183 019. The IG shall be responsible for resolving the domain name.	IETF RFC 3261 REGISTER <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact Remarks: a) The contact shall include the Feature Tags parameter. b) The URI parameter shall be included. c) The expires parameter should be included. d) The priority parameter should be included IG adds all the other mandatory parameters that are absent in the X-OITF-Contact. Default values are assigned by the IG to OPTIONAL parameters that are not provided in the X-OITF-Contact. e) Thesip instance feature tag shall be included according to 4.1 and 4.2 of IETF RFC 5626. Its format shall be identical to the format specified in IETF RFC 4122. The sip instance feature tag shall have the following syntax: +sip.instance="<urn:uuid:Unique-Instance>" where Unique-Instance shall be 128 bit and shall conform to the syntax in IETF RFC 4122. The Node field is a 48 bit field which shall be populated with the first 48 bit from the value employed in <deviceId> used at restart or powerup of an OITF device. The sip instance feature tag shall be persistent across power cycles of the device. All OITFs that want to be able to use the session transfer feature shall register the g.3gpp.icsi-ref media feature tag containing the IPTV IMS communication service identifier. In particular the X-OITF-Contact header shall have the following media feature tag included: g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.iptv"	IETF RFC 3261 and IETF RFC 3840
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Supported set to "gruu" if the feature is required	IETF RFC 3261 IETF RFC 5627

Table 61 –HTTP extension headers for user registration/de-registration response (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	SIP header field prefixed with X-OITF
X-OITF-To	SIP header field prefixed with X-OITF
X-OITF-Expires	SIP header field prefixed with X-OITF
X-OITF-Contact The returned contact shall include the following 3 elements if the GRUU feature is requested by the OITF: pub-gruu, temp-gruu and the sip instance feature tag	SIP header field prefixed with X-OITF IETF RFC 5627
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

If the OITF does not support native HNI-IGI, user deregistration shall be done through a DAE application.

6.4.6.1.3 Procedure for refreshing a registration

This procedure may be initiated by the OITF at any time before the expiry of the registration refresh timer.

The procedure is the same as the procedure for registering a user. A registration shall be terminated if it is not refreshed before the expiry of the registration refresh timer.

For an OITF-initiated registration, the IG shall consider a registration terminated (that is, the user de-registered) if it is not refreshed. In this case, the IG executed the procedures associated with user deregistration.

6.4.6.1.4 Procedure for subscription to the registration event package

This procedure shall be invoked immediately after the successful registration of an IMPU (including the default identity) or an IPTV end-user identity.

Step 1: The OITF shall send an HTTP POST request to the IG on the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 62.

HTTP request body: empty

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the outgoing subscription request message, as per Table 62. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for the rejection.

Step 3: The IG shall send a SIP SUBSCRIBE to the network, to subscribe to the Registration event, and shall wait for the response to the subscription request. The IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the subscription request. The response shall include a list of SIP headers as per Table 63 in addition to the normal HTTP headers as per IETF RFC 2616.

- Step 4:** Following that, the OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any response.
- Step 5:** When a SIP NOTIFY is received by the IG, the IG shall return a HTTP 200 OK response to the OITF. The response shall include the list of SIP headers as per Table 64 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the SIP body received in the incoming NOTIFY (See also 7.2.4.2.2) authentication.
- Step 6:** Once the OITF accepts the incoming SIP NOTIFY, it shall send an HTTP POST PENDING_IG request to the IG. The content of the HTTP request shall be as follows.
- HTTP request header: It includes the following:
- <list of HTTP headers> – as per IETF RFC 2616;
 - <list of SIP headers encoded as HTTP headers> – as per Table 65
- HTTP request body: empty
- Step 7:** The IG shall send the SIP 200 OK response to the network and then shall return to Step 5 to handle any subsequent NOTIFY received from the network.

Table 62 – Supported HTTP extension headers in the HNI-IGI SUBSCRIBE request for the registration event package

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI shall be set to the Public identity of the IPTV end user who has just registered.	IETF RFC 3261 SUBSCRIBE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3265 and IETF RFC 3680 (registration event)
X-OITF-Accept	IETF RFC 3265 and IETF RFC 3680
X-OITF-Contact Remarks: a) URI parameter shall be included, and shall match what is sent in the Contact header included in the registration request. b) Expires parameter should be included. c) Priority parameter should be included. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261

Table 63 – Supported HTTP extension headers in the response to an HNI-IGI SUBSCRIBE request for the registration event package

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 64 –HTTP extension headers for a HNI-IGI NOTIFY request sent IG→OITF

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP SUBSCRIBE	IETF RFC 3261 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3265 and IETF RFC 3680
X-OITF-Call-ID	IETF RFC 3265 and IETF RFC 3680
X-OITF-Subscription-State	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3265 and IETF RFC 3680
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 65 –HTTP extension headers in the response to a NOTIFY request

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

6.4.6.1.5 Procedure for terminating a subscription to the registration event package

This procedure shall be invoked prior to de-registering a user.

The procedure is the same as the procedure for initiating a subscription to the Registration event, however in this case the X-OITF-Expires header in Table 62 shall be set to 0.

For an OITF-initiated registration, the IG shall consider a subscription terminated if is not refreshed.

6.4.6.1.6 Refreshing subscription to registration event

The procedure is the same as the procedure for initiating a subscription.

It is the responsibility of the application initiating the subscription procedure to refresh the subscription according to the refresh subscription timer information received in the response to the subscription request. Refreshing the subscription should be performed before the expiry of the refresh timer. A subscription that is not refreshed before the expiration of the refresh timer shall be terminated.

6.4.6.1.7 Registration of DAE/embedded applications

IMS applications, DAE or embedded, that are initiated in the OITF and expect unsolicited incoming messages shall register with the IMS network the feature tags and/or the appropriate service URN (ICSI) and /or IMS application reference identifier (IARI) for the initiated application where mandated by the specification governing the application ETSI TS 183 063, OMA-TS-SIMPLE_IM-V1_0-20100322-C, ETSI TS 124 503, IETF RFC 3840, IETF RFC 3841. This allows unsolicited incoming SIP messages destined for users and targeted for these applications to be delivered to the appropriate application instance in the OITF.

The procedure used by an application for registering the appropriate feature tags and/or service URN (ICSI) and/or IARI is the same procedure used for user registration.

6.4.6.2 GBA authentication

6.4.6.2.1 General

Note that GBA authentication can be achieved using either the GBA authentication using IMS Gateway procedure, specified in IEC 62766-7:2017, 5.4.5 or the more general procedure, HTTP Digest authentication using IMS Gateway in IEC 62766-7:2017, 5.4.4. The latter, more general procedure, allows the use of different authentication mechanism in a way that is transparent to the OITF, including possible future authentication mechanisms, and should preferably be used. It is expected that GBA authentication using IMS Gateway procedure will be deprecated and removed in future versions of this specification.

This subclause describes the HNI-IGI message for the GBA authentication. For the details of the sequence for GBA authentication, refer to subclause 5.4.4 of IEC 62766-7:2017. Note that GBA authentication applies only for user registration and authentication based on IMS AKA.

6.4.6.2.2 Initial GBA registration

After IMS registration is successfully performed, and if the IG supports GBA authentication, OITFs supporting native HNI-IGI shall issue the following GBA registration request to the IG. OITFs that do not support native HNI-IGI do not support GBA.

Step 1: The OITF shall send an HTTP POST request to the IG. The content of the HTTP request shall be as follows.

HTTP request headers: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI-request: GBA-Registration>

HTTP request body: empty

Step 2: After the GBA bootstrapping procedure over UNIS-9, the IG returns an HTTP 200 OK response with an empty body.

6.4.6.2.3 Credential retrieval by an OITF for re-use of GBA authentication

The key Ks that is established during the GBA registration may be reused later for user authentication and service access by consumer network applications.

Each time an OITF needs to access a service that is offered by an AS (i.e. NAF) that requires GBA authentication, a specific key Ks_NAF or Ks_ext_NAF shall be derived respectively by the IG or by the ISIM in the IG and the server side GBA Single Sign-on function (the BSF). This generated key shall be conveyed to the OITF in the consumer network by the IG, and to the AS by the server side GBA Single Sign-on function (the BSF). The key Ks_NAF or Ks_ext_NAF shall then be used for authentication between the OITF and the AS, using HTTP Digest authentication as specified by 3GPP TS 24.109. The OITF shall act as the OIPF as specified in 3GPP TS 24.109.

As a pre-requisite to this procedure, the GBA procedure shall have been successfully completed.

The complete procedure for retrieval of credentials by the OITF from the IG is specified in IEC 62766-7.

The HNI-IGI procedure for credential retrieval is as follows:

Step 1: The OITF shall send an HTTP POST request to the IG. The request includes the FQDN of the NAF. The content of the HTTP request shall be as follows.

HTTP request headers: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI-request> – set to Fetch-GBA-Credentials
- <X-HNI-IGI-NAF-FQDN> – set to NAF FQDN extracted from the HTTP authentication realm as specified in 3GPP TS 24.109.

HTTP request body: Empty

Step 2: The IG shall generate Ks_NAF or Ks_ext_NAF with the ISIM. For clarity this specific key is named in the rest of the document Ks_(ext)_NAF and will refer to Ks_NAF in case of GBA_ME and Ks_ext_NAF in case of GBA_U and is computed as follows:

$Ks_{(ext)}_{NAF} = KDF(Ks, "gba-me", RAND, IMPI, NAF_ID)$, where KDF is the key derivation function as specified in Annex B of 3GPP TS 33.220 and the key derivation parameters consist of the user's IMPI, the NAF_ID and RAND. The NAF_ID is constructed as follows: $NAF_ID = FQDN\ of\ the\ NAF || Ua\ security\ protocol\ identifier$ as specified in 3GPP TS 33.220. The identifier for Ua security protocol HTTP Digest authentication according to 3GPP TS 33.220 is (0x01,0x00,0x00, 0x00,0x02).

The IG shall return an HTTP 200 OK to the OITF that includes the Ks_(ext)_NAF, the B-TID, the lifetime of the key Ks_(ext)_NAF and OPTIONALLY the intended identity. The lifetime indicates the expiry time of the key Ks_(ext)_NAF and is equal to the lifetime of the key Ks (which was specified by the BSF during the GBA bootstrapping procedure). The content of the HTTP 200 OK response is as follows.

HTTP response headers: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI-KS_NAF> – set to the computed Ks_(ext)_NAF
- <X-HNI-IGI-B_TID> – set to the B-TID
- <X-HNI-IGI-LifeTime> – set to life time of the key Ks_NAF

- <X-HNI-IGI-Intended-Identity> – set to the intended identity. This header is OPTIONAL and its use is described in IEC 62766-7.

HTTP response body: empty

6.4.6.3 HTTP digest authentication

6.4.6.3.1 General

This subclause describes the HNI-IGI messages for the HTTP Digest authentication using IG. For the details of the sequence for HTTP Digest authentication using IG, refer to 5.4.4 of IEC 62766-7:2017.

6.4.6.3.2 HTTP realm retrieval by an OITF

After IMS registration is successfully performed, and if the IG supports HTTP digest authentication, OITFs supporting native HNI-IGI shall issue an HTTP realm retrieval request to the IG. OITFs that do not support native HNI-IGI do not support HTTP digest authentication using IG.

Step 1: The OITF shall send an HTTP POST request to the IG. The content of the HTTP request shall be as follows:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI-request> – set to Fetch-HTTP-Realms

HTTP request body: Empty

Step 2: The IG shall return an HTTP 200 OK to the OITF that includes the list of supported auth-scheme and realm. The content of the HTTP 200 OK response shall be as follows:

HTTP response headers: It includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI-Auth-Realms> – set to a list of comma-separated realm-value, for which HTTP authentication credentials are available in the IG. Realm-values are quoted string as defined in IETF RFC 2617.
- <X-HNI-IGI-User-Agent-Tokens> – set to a list of comma-separated tokens to append to the HTTP User-Agent of the OITF for signalling support of specific authentication schemes. If the IG supports GBA authentication, the IG shall add “3gpp-gba” to the returned User-Agent tokens.

HTTP response body: empty

6.4.6.3.3 HTTP credential retrieval by an OITF

If the OITF has registered to an IG which supports HTTP digest authentication using IG, each time the OITF needs to access a service offered by an application server that requires HTTP authentication, the OITF may use credentials retrieved from the IG. The conditions under which an OITF uses HTTP credentials retrieved from the IG are described in IEC 62766-7.

The complete procedure for use of HTTP credentials by the OITF retrieved from the IG is specified in IEC 62766-7.

The HNI-IGI procedure for HTTP credential retrieval is as follows:

Step 1: The OITF shall send an HTTP POST request to the IG. The request includes the auth-scheme and realm as defined in IETF RFC 2617. The content of the HTTP request shall be as follows.

HTTP request headers: including the following:

- <list of HTTP headers> – as per IETF RFC 2616;

- <X-HNI-IGI-request> – set to Fetch-HTTP-Credentials;
- <X-HNI-IGI-User> – set to the IMPU of the currently registered user;
- <X-HNI-IGI-Auth-Scheme> – set to auth-scheme as defined in IETF RFC 2617;
- <X-HNI-IGI-Auth-Realm> – set to realm as defined in IETF RFC 2617;
- <X-HNI-IGI-Auth-Nonce> – set to nonce (server nonce) as defined in IETF RFC 2617;
- <X-HNI-IGI-Auth-CNonce> – set to cnonce (client nonce) as defined in IETF RFC 2617;
- <X-HNI-IGI-Auth-Algorithm> – set to algorithm as defined in IETF RFC 2617.

HTTP request body: empty

Step 2: The IG shall return an HTTP 200 OK to the OITF that includes the user-id and password for the given auth-scheme and realm. The content of the HTTP 200 OK response is as follows:

HTTP response headers: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616.
- <X-HNI-IGI-Auth-Credentials> – this header is set to H(A1) as defined in IETF RFC 2617. If this header is empty or not present, there are no credentials available in the IG for this request.
- <X-HNI-IGI-Intended-Identity> – set to the intended identity. This header is optional and its use is described in IEC 62766-7.

HTTP response body: empty.

6.4.6.4 User ID retrieval for services relying on IMS credentials

The OITF shall retrieve a list of user IDs (IMPU and Alias) from the IG for service over the HNI-IGI interface. This procedure should not require user authentication. The IG shall at a minimum provide the default identity for the household and may provide all available identities. OITFs that do not support native HNI-IGI should retrieve the User IDs using DAE.

Step 1: The OITF shall send an HTTP POST request to the IG. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <X-HNI-IGI-request: Fetch-UserIDs>.

HTTP request body: empty

Step 2: The IG returns a list of user IDs (IMPU and Aliases) as follows:

The IG shall return an HTTP 200 OK to the OITF. The content of the HTTP 200 OK response shall be as follows:

HTTP response headers: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616

HTTP response body: a list of IMPUs and Display names (alias). Elements are separated with commas, entries are separated with semi-colon, in the format <IMPU1>,<alias1>;<IMPU2>,<alias2>;... etc. The first entry shall be the default identity for the household.

The usage of IMPU and alias by the OITF is defined by the CSP specification.

Depending on the policy of the IG and service provider, the IG may return the default identity only. In this case, the user of the OITF shall be required to enter a user ID manually.

6.5 Protocols for communications functions using SIP

6.5.1 CallerID

6.5.1.1 Procedure for instant message based caller ID – HTTP option Procedure on HNI-IGI

The OITF supports the following procedure for caller ID. The incoming message carrying a Caller ID can either be handled by a native application in the OITF, or in a DAE application. The same HNI-IGI message format is used in either case.

- Step 1:** The IG receives an incoming SIP MESSAGE from the network.
- Step 2:** The IG forwards the information in the SIP MESSAGE to the OITF in the HTTP 200 OK response to a PENDING_IG request that was established when the application started. The list of SIP headers to be included in the message to the OITF shall be as per Table 66. The body of the SIP MESSAGE shall be included in the HTTP response body.
- Step 3:** Upon receipt of the message, the OITF shall issue an HTTP POST request. The content of the HTTP request shall be as follows:
- HTTP request header: including the following:
- <list of HTTP headers> – as per IETF RFC 2616;
 - <list of SIP headers encoded as HTTP headers> – as per Table 67.
- HTTP request body: empty
- Step 4:** The IG shall send SIP 200 OK to the network.

NOTE For handling of new incoming SIP MESSAGE, refer to 5.3.2 of IEC 62766-5-1:2017.

Table 66 –HTTP extension headers for an instant message based caller ID (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the IMS Public Identity (IMPU) of the target of the message	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3428, Draft OMA-TS-SIMPLE_IM-V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-OITF-Content-Length	IETF RFC 3261

Table 67 –HTTP extension headers for the response to an Instant message based caller ID (OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.5.1.2 Procedure for IMS telephony based caller ID (optional) – Procedure for HNI-IGI – HTTP option

The following procedure may be supported in the OITF for caller ID presentation to the OITF user as a result for an incoming IMS voice call to the IG.

The incoming message, carrying information on the IMS voice call, can either be handled by a native application in the OITF, or by a DAE application. The same HNI-IGI message format is used in either case.

- Step 1:** The IG receives an incoming SIP INVITE.
- Step 2:** The IG forwards the SIP INVITE to the OITF as an HTTP response to a PENDING_IG request. The list of SIP headers to be included in the message to the OITF shall be as per Table 68. The content of the invite message shall also be included.
- Step 3:** Upon receipt of the message, the OITF issues an HTTP POST request indicating that the voice call is not supported by the OITF by response code 415 Unsupported Media Type. Other values may be used according to IETF RFC 3261. The content of the HTTP request is as follows.
- HTTP request header including the following:
- <list of HTTP headers> – as per IETF RFC 2616;
 - <list of SIP headers encoded as HTTP headers> – as per Table 69.
- HTTP request body: application/sdp
- Step 4:** The IG shall forward the SIP response to the network.
- Step 5:** When the IG receives the SIP ACK from the network and shall forward it to the OITF as an HTTP response to a PENDING_IG request. The list of SIP headers to be included in the message to the OITF shall be as per Table 70.

NOTE For handling of new incoming INVITE messages for new dialogs, refer to 5.3.2 of the DAE specification IEC 62766-5-1:2017.

Table 68 –HTTP extension headers on the HNI-IGI interface (IG→OITF) for a received SIP INVITE

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the IMS Public User Identity of the target of the message.	IETF RFC 3261 INVITE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line"	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261, ES 283 002
X-OITF-P-Called-Party-ID	ETSI ES 283 002
X-OITF-P-Asserted-Identity	ETSI ES 283 002

Table 69 –HTTP extension headers on the HNI-IGI interface (OITF→IG) for a response to the SIP INVITE

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Accept	IETF RFC 3261

Table 70 –HTTP headers in the HNI-IGI ACK message (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line	IETF RFC 3261 ACK <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Accept	IETF RFC 3261

6.5.2 Instant messaging

Procedure for instant messaging on HGI-INI – HTTP option

6.5.2.1 General

Instant messaging on the OITF uses the HNI-IGI functionality, as described in 6.6.

There are two cases, messages originating from the OITF, and messages terminating in the OITF.

6.5.2.2 Procedure for OITF originating an instant message

The following procedure is supported in the OITF to originate instant messages:

An instant message can either originate from a native application in the OITF or from a DAE application. The same HNI-IGI message format is used.

Step 1: The OITF shall send an HTTP POST request to the IG using the HNI-IGI functionality, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 71.

HTTP request body: the content type as per IETF RFC 3428.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the message as per Table 71. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall send a SIP MESSAGE to the network. When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the SIP MESSAGE. The response includes a list of SIP headers as per Table 72 in addition to the normal HTTP headers as per IETF RFC 2616.

Table 71 –HTTP extension headers for an outgoing instant message (OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the Public identity of the target of the message.	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line".	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

Table 72 –HTTP extension headers for the response to an outgoing and incoming Instant message (IG→OITF and OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.5.2.3 Incoming instant messaging procedure

The following procedure is supported in the OITF for incoming instant messages.

The incoming message can be handled either by a native application in the OITF, or in a DAE application. The same HNI-IGI message format is used in either case.

Step 1: The IG receives an incoming SIP MESSAGE.

Step 2: The IG shall forward the SIP MESSAGE to the OITF as an HTTP response to a PENDING_IG request. The list of SIP headers to be included in the notification forwarded to the OITF shall be as per Table 73. The body of the SIP MESSAGE shall be included in the HTTP body.

Step 3: Upon receipt of the message, the OITF shall issue an HTTP POST request. The content of the HTTP request shall be as follows.

HTTP request header. It includes the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 72.

HTTP request body: empty

Step 4: The IG shall forward the SIP 200 OK to the network

Table 73 – List of HTTP extension headers for an incoming instant message (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the Public identity of the target of the message.	IETF RFC 3261 MESSAGE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line".	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Type	IETF RFC 3261, Draft OMA-TS-SIMPLE_IM-V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-OITF-Content-Length	IETF RFC 3261

6.5.3 IM session (chat using MSRP)

6.5.3.1 Procedure for initiating an instant messaging session (MSRP chat) – HTTP option

To initiate a chatting session using MSRP, the OITF shall use the following procedure.

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.

HTTP request header. including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 74.

HTTP request body: empty

Step 2: If the request is for an instant messaging MSRP chat session, the IG shall validate that the request includes all the mandatory SIP headers as per Table 74. The IG shall reject a request that is missing any mandatory SIP headers with a non-200

OK HTTP response, including the reason for rejection. The IG shall generate the INVITE by mapping the X-OITF headers to the appropriate SIP header. As the IG implements MSRP, the IG shall include all the necessary additional SIP headers and the SDP body to initiate the MSRP session as follows:

- The Content-Type header shall be added and set to “application/sdp”
- The Content-Length header shall be added and set to the appropriate value
- The message body shall include the following information:
 - A, c = IN IP4 <IP address>, where <IP address> would contain the IP address of the IG;
 - An, m = message <tcp port> tcp/msrp, where tcp port is a TCP port could be set to the dummy value “9”;
 - An, a = accept-types:message/cpim, attribute which is mapped from the “X-OITF-Accept:” header value;
 - An a = path msrp://<IP address>:<tcpport>/<session-id>; tcp, where
 - <IP address> would contain the IP address of the IG;
 - <tcpport> would be assigned automatically by the IG;
 - <session-id> would be assigned automatically by the IG and bound to the requesting OITF chatting application.

In this case the IG is not service agnostic. The IG detects that this session is for MSRP by examining the X-OITF-Accept header which shall include message/cpim (see example in A.2.3, “chat”).

Step 3: The IG shall send a HTTP 200 OK response to the OITF when the SIP 200 OK is received as a response to the session invitation. The SIP 200 OK headers are mapped as indicated in Table 75, in addition to the normal HTTP 200 OK headers. The IG shall not forward the body of the SIP 200 OK to the OITF. The IG shall establish and maintain the MSRP state information including the binding between the logical entities (indicated in the From and To headers) and the corresponding path (the one initiated by the IG for the OITF and the one indicated by the distant entity for the To:). The IG shall maintain a binding between the SIP dialog and the MSRP state information for the duration of the SIP dialog.

Step 4: Upon receipt of a 200 OK response, the OITF shall send an HTTP PENDING_IG to acknowledge the final response. The content of the HTTP request shall be as follows.

HTTP request header, including the following:

- <list of HTTP headers> – as per IETF RFC 2616;
- <list of SIP headers encoded as HTTP headers> – as per Table 76.

HTTP request body: empty

Table 74 –HTTP extension headers for IM INVITE request (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request URI shall be set to the IMPU of the subscriber with whom the session is requested.	IETF RFC 3261 INVITE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line INVITE" header.	IETF RFC 3261
X-OITF-Contact Remarks: a) URI parameter shall be included and shall match what is sent in the Contact header included in the registration request. b) Expires parameter should be included.	IETF RFC 3261
X-OITF-Accept-Contact	Set according to OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Accept shall be set to: "message/cpim"	OMA-TS-SIMPLE_IM-V1_0-20100322-C

Table 75 –HTTP extension headers for a 200 OK response received for the INVITE IG→OITF

X-OITF HTTP header	for coding purposes
X-OITF-response-Line	RFC 3261 SIP/2.0 <response>
X-OITF-From	RFC 3261
X-OITF-To	RFC 3261
X-OITF-Call-ID	RFC 3261
X-OITF-CSeq	RFC 3261
X-OITF-Contact	RFC 3261
X-OITF-Accept	RFC 3261

Table 76 – List of HTTP extension headers in HNI-IGI ACK request

X-OITF HTTP header	Source of Information for coding purposes
X-OITF-request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message	RFC 3261 ACK <request URI> SIP/2.0
X-OITF-From	RFC 3261
X-OITF-To	RFCIETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what has been inserted in the INVITE message. IG includes all other mandatory parameters that are absent.	IETF RFC 3261

6.5.3.2 MSRP invocation – HTTP option

6.5.3.2.1 General

The OITF shall access MSRP capabilities in the IG using the X-HNI-IGI headers.

6.5.3.2.2 Outgoing MSRP chat messages

The OITF shall send an outgoing MSRP chat message using the following procedure:

- Step 1:** The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:
- HTTP request header: including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of HNI-IGI headers encoded as HTTP headers> – as per Table 77
- HTTP request body: the message in plain text.
- Step 2:** The IG shall validate that the request includes all the mandatory HNI-IGI headers for the process as per Table 77. The IG shall reject a request that is missing any mandatory HNI-IGI headers with a non-200 OK HTTP response, including the reason for rejection.
- Step 3:** The IG shall validate the Call-ID and the message-ID, if present, and subsequently shall send an MSRP SEND message to the network, then wait for the MSRP 200 OK response from the network. The IG shall return a HTTP 200 OK response to the OITF when it receives the MSRP 200 OK (or other responses). The HTTP 200 OK response shall include the HNI-IGI headers as per Table 78 in addition to the normal HTTP headers as per IETF RFC 2616.

Table 77 –HNI-IGI HTTP extension headers for an MSRP SEND request (OITF→IG)

X-HNI-IGI HTTP header	Source of information for coding purposes
X-HNI-IGI-request SEND MESSAGE	OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-message-ID	shall be left blank for the first message.
X-HNI-IGI-Call-ID	shall be set to the same value for the INVITE transaction that initiated the session
X-HNI-IGI-From	shall be set to the identity of the originator of the message
X-HNI-IGI-To	shall be set to the identity of the recipient of the message

Table 78 –HNI-IGI HTTP extension headers included in the HTTP 200 OK response (IG→OITF)

X-HNI-IGI headers	Source of information for coding purposes
X-HNI-IGI-response MSRP <response>	OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-message-ID	OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-From	OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-To	OMA-TS-SIMPLE_IM-V1_0-20100322-C

6.5.3.2.3 Sending an MSRP chat state message

The OITF shall use the following procedure to indicate the activity of the user (e.g., “is composing”):

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header: Including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of HNI- IGI headers encoded as HTTP headers> – as per Table 79

HTTP request body: shall contain the appropriate XML document as indicated in IETF RFC 3994 and OMA-TS-SIMPLE-IM_V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C.

Step 2: The IG shall validate that the request includes all the mandatory HNI-IGI headers for the process as per Table 79. The IG shall reject a request that is missing any mandatory HNI-IGI headers with a non-200 OK HTTP response, including the reason for rejection.

Step 3: The IG shall validate the Call-ID and the message-ID, and send an MSRP SEND message to the network after performing the necessary mapping and adding the appropriate tags. The IG shall then wait for the MSRP 200 OK response from the network. The IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF when it receives the MSRP 200 OK (or other responses). The response shall include a list of HNI-IGI headers as per Table 78 in addition to the normal HTTP headers as per IETF RFC 2616.

Table 79 –HNI-IGI HTTP extension headers for an MSRP SEND ACTIVITY request (OITF→IG)

X-HNI-IGI HTTP header	Source of information for coding purposes
X-HNI-IGI-request MSRP SEND ACTIVITY	OMA-TS-SIMPLE_IM-V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-message-ID	shall be set to the appropriate message id
X-HNI-IGI-Call-ID	shall be set to the same value for the INVITE transaction that initiated the session
X-HNI-IGI-From	OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-To	OMA-TS-SIMPLE_IM-V1_0-20100322-C

6.5.3.2.4 Receiving an MSRP chat message

The IG shall use the following procedure when receiving an incoming MSRP message:

- Step 1:** In response to a PENDING_IG request, the IG shall send an HTTP 200 OK response to the OITF over the HNI-IGI interface, as described in 6.6. The response shall include the HNI-IGI headers listed in Table 80, in addition to the mandatory HTTP headers in IETF RFC 2616. The body of the HTTP 200 OK response shall include the received text.
- Step 2:** The OITF shall respond with an HTTP POST request with its body containing an MSRP 200 OK response. The contents of the HTTP request shall be as follows:
HTTP request header: including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of HNI-IGI headers encoded in HTTP headers> – as per Table 81
- HTTP request body: empty
- Step 3:** The IG shall send the response from the OITF in an MSRP response message to the network after performing the necessary validation.

Table 80 –HNI-IGI HTTP extension headers for an incoming MSRP message (IG→OITF)

X-HNI-IGI HTTP header	Source of information for coding purposes
X-HNI-IGI-request MSRP RECEIVE MESSAGE	OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-message-ID	shall be set to the appropriate message id
X-HNI-IGI-Call-ID	shall be set to the same value for the INVITE transaction that initiated the session
X-HNI-IGI-From	shall be set to the remote user
X-HNI-IGI-To	shall be set to the recipient of the message

Table 81 –HNI-IGI HTTP extension headers for an MSRP 200 OK response to an incoming MSRP message (OITF→IG)

X-HNI-IGI HTTP header	Source of information for coding purposes
X-HNI-IGI-response MSRP <response>	Set to the appropriate response
X-HNI-IGI-message-ID	shall be set to the appropriate message id

6.5.3.2.5 Receiving an MSRP chat state message

The IG shall use the following procedure when receiving an incoming MSRP chat state message:

- Step 1:** In response to a PENDING_IG request, the IG shall send an HTTP 200 OK response to the OITF over the HNI-IGI interface, as described in OITF-IG Interface (HNI-IGI). The response shall include the HNI-IGI headers listed in Table 82 in addition to the mandatory HTTP headers in IETF RFC 2616. The body of the HTTP 200 OK response shall contain the appropriate XML document as indicated in IETF RFC 3994 and OMA-TS-SIMPLE-IM_V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C.
- Step 2:** The OITF shall respond with an HTTP POST request with its body containing an MSRP 200 OK response. The contents of the HTTP request shall be as follows.
HTTP request header: including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of HNI-IGI headers encoded in HTTP headers> – as per Table 81
- HTTP request body: empty
- Step 3:** The IG shall send the response from the OITF in an MSRP response message to the network after performing the necessary validation.

**Table 82 –HNI-IGI HTTP extension headers
for an incoming MSRP receive activity (IG→OITF)**

X-HNI-IGI HTTP header	Source of information for coding purposes
X-HNI-IGI-request MSRP RECEIVE ACTIVITY	OMA-TS-SIMPLE-IM_V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-HNI-IGI-message-ID	shall be set to the appropriate message id
X-HNI-IGI-Call-ID	shall be set to the same value for the INVITE transaction that initiated the session
X-HNI-IGI-From	shall be set to the remote user
X-HNI-IGI-To	shall be set to the recipient of the message

6.5.3.3 Terminating an IM session (MSRP chat) – HTTP option

In order to terminate an MSRP session, the OITF shall use the following procedure:

- Step 1:** The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.
HTTP request header: including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 83
- HTTP request body: empty
- Step 2:** The IG shall validate that the request includes all the mandatory SIP headers for the process as per Table 83. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection. The IG shall generate the SIP BYE by mapping the X-OITF headers to the appropriate SIP headers.
- Step 3:** The IG shall send a HTTP 200 OK response to the OITF when the SIP 200 OK is received as a response to the chat session termination request. The SIP 200 OK headers are mapped as indicated in Table 84 in addition to the normal HTTP 200 OK headers.

Table 83 –HTTP extension headers for an MSRP BYE request (OITF→IG)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line	IETF RFC 3261 BYE <request URI> SIP/ 2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact shall be set to the value received in the contact of a 200 OK for session termination or SIP INVITE for session origination	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Length Shall be set to 0	IETF RFC 3261

Table 84 –HTTP extension headers for a 200 OK response to a BYE (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 200 OK
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Content-Length Set to 0	IETF RFC 3261

6.5.3.4 Remote termination of an IM session (MSRP chat) – HTTP option

The IG shall use the following procedure when receiving an incoming SIP BYE message for an ongoing IM session (MSRP chat):

Step 1: The IG receives a SIP BYE message from the network.

Step 2: The IG forwards the information in the SIP BYE to the OITF over the HNI-IGI interface in the HTTP 200 OK response to a PENDING_IG request. The response shall include the list of SIP headers listed in Table 85, in addition to the mandatory HTTP headers in IETF RFC 2616.

Step 3: The OITF shall respond with an HTTP POST request. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded in HTTP headers> – as per Table 86

HTTP request body: empty

Step 4: The IG shall send SIP 200 OK to the network.

**Table 85 –HTTP extension headers
for an incoming SIP BYE (IG→OITF)**

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP INVITE (for outgoing session) or a 200 OK (for incoming session)	IETF RFC 3261 BYE <request URI> SIP/ 2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Length Shall be set to 0	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

**Table 86 –HTTP extension headers for
the response to an SIP BYE (OITF→IG)**

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.5.3.5 Procedure for reception of a remotely initiated instant messaging session (MSRP chat) – HTTP option

The IG shall use the following procedure when receiving an incoming SIP INVITE message for a new IM session (MSRP chat):

Step 1: The IG receives a SIP INVITE message from the network.

Step 2: The IG shall validate that the request includes all the mandatory SIP headers as per Table 87. This is required since the IG shall send all this information to the OITF. The IG shall reject any incoming request that is missing any mandatory parameter. Subsequently, the IG shall perform the following checks:

- Ensure that the Content-Type header is present and set to “application/sdp”
- Verify that the SDP body includes the following information:
 - A c = IN IP4 <IP address>, where <IP address> would contain the remote IP address.
 - An m = message <tcp port> tcp/msrp, where tcp port is a TCP port and could be set to the dummy value “9”
 - An a = accept-types:message/cpim, attribute which is mapped from the Accept header value.
 - A a = path msrp://<IP address>:<tcpport>/<session-id>; tcp, where:
 - <IP address> would contain the remote IP address
 - <tcpport> remote IP port
 - <session-id> assigned automatically by the remote peer.

Step 3: Following that, the IG retains and stores information in the SDP, and forwards only the information in the SIP INVITE headers to the OITF over the HNI-IGI interface in the HTTP 200 OK response to a PENDING_IG request that was sent by the OITF to the IG when the application was launched. The response shall include the list of SIP headers listed in Table 87, in addition to the mandatory HTTP headers in IETF RFC 2616.

Step 4: The OITF shall respond with an HTTP POST request that includes the OITF response to the incoming INVITE. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded in HTTP headers> – as per Table 88

HTTP request body: empty

Step 5: The IG shall append the SDP to the SIP 200 OK before sending it to the network. The appended SDP shall include the following information:

- A c = IN IP4 <IP address>, where <IP address> would contain the IP address of the IG,
- An m = message <tcp port> tcp/msrp, where tcp port is a TCP port could be set to the dummy value “9”
- An a = accept-types:message/cpim, attribute which is mapped from the “X-OITF-Accept:” header value
- An a = path msrp://<IP address>:<tcpport>/<session-id>; tcp, where:
 - <IP address> would contain the IP address of the IG
 - <tcpport> would be assigned automatically by the IG
 - <session-id> would be assigned automatically by the IG and bound to the responding OITF chatting application

Step 6: The IG receives a SIP ACK message from the network

Step 7: Following that, the IG shall send the information in the incoming ACK message to the OITF in a HTTP 200 OK response. The response includes a list of SIP headers as per Table 89.

NOTE Any SDP information is retained in the IG since the IG handles the MSPP protocol.

Step 8: The OITF shall send an HTTP HNI-IGI PENDING_IG request to the IG and shall wait for any incoming messages.

**Table 87 –HTTP extension headers for
an incoming IM INVITE request (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request URI shall be set to the IMPU of the subscriber with whom the session is intended	IETF RFC 3261 INVITE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To shall be set to the value of the request URI in the “X-OITF-request-Line INVITE” header	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261
X-OITF-Accept-Contact	Set according to OMA-TS-SIMPLE_IM-V1_0-20100322-C
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Accept shall be set to: “message/cpim”	OMA-TS-SIMPLE_IM-V1_0-20100322-C

**Table 88 – List of HTTP extension headers for the response
to an incoming IM INVITE request (OITF→IG)**

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Accept shall be set to “message/cpim”	IETF RFC 3261
X-OITF-Contact Remarks: a) URI parameter shall be included and shall match what is returned in the contact header includes in the response to the registration process. b) Expires parameter should be included.	IETF RFC 3261

Table 89 – Supported HTTP extension headers in HNI-IGI ACK request for successful IM session (MSRP chat) (IG→OITF)

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message.	IETF RFC 3261 ACK <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line".	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what has been received in the incoming INVITE message.	IETF RFC 3261

6.5.4 Presence

6.5.4.1 Procedures for subscription to presence on the HNI-IGI interface – HTTP option

The procedure for subscription to the presence event shall be invoked from either a DAE application or an embedded application in the OITF. The procedure is as follows:

- Step 1:** The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.
- HTTP request header including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 90
- HTTP request body: empty
- Step 2:** The IG shall validate that the request includes all the mandatory SIP headers for the subscription process as per Table 90. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.
- Step 3:** The IG shall send a SIP SUBSCRIBE to the network, to subscribe to the presence event, and shall wait for the response to the subscription request. The IG shall then return a HTTP 200 OK response to the OITF to report the response to the subscription request. The response includes a list of SIP headers as per Table 91, in addition to the normal HTTP headers as per IETF RFC 2616.
- Step 4:** The OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to subclause 6.6.1) and shall wait for any incoming messages.
- Step 5:** When a SIP NOTIFY is received by the IG, the IG shall return a HTTP 200 OK response to the OITF containing the information in the incoming NOTIFY message. The response includes a list of SIP headers as per Table 92 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the SIP body received in the incoming NOTIFY compliant to Annex E of ETSI TS 183 063.
- Step 6:** Once the OITF accepts the incoming SIP NOTIFY, it shall send an HTTP POST PENDING_IG request to the IG to acknowledge the receipt of notification and issue a new pending HTTP request. The content of the HTTP request shall be as follows:

HTTP request header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 93

HTTP request body: empty

Step 7: The IG shall send the SIP 200 OK response to the network and then shall return to Step 5 to handle any subsequent NOTIFY received from the network.

The OITF shall ensure that the presence related data conforms to the appropriate XML schemas.

6.5.4.2 Procedure for cancellation of a subscription to presence on the HNI-IGI interface – HTTP option

This procedure may be invoked at any time.

The OITF shall de-register the IPTV end user before invoking this procedure.

The procedure is essentially the same as the procedure for initiating a subscription to the presence event, except that the X-OITF-Expires header in Table 90 shall be set to 0.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

**Table 90 –HTTP extension headers
for a SUBSCRIBE request (OITF→IG)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI shall be set to the Public identity of the IPTV end user who has just registered.	IETF RFC 3261 SUBSCRIBE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3621
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line".	IETF RFC 3621
X-OITF-Event	IETF RFC 3265, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-Accept	IETF RFC 3265, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-Contact Remarks: a) URI parameter shall be included, and shall match the Contact header included in the registration request. b) Expires parameter should be included. c) Priority parameter should be included. IG includes all other mandatory parameters that are absent.	IETF RFC 3621
X-OITF-Call-ID	IETF RFC 3621
X-OITF-CSeq	IETF RFC 3621
X-OITF-Expires Note that if absent a default value shall be assumed by the IG.	IETF RFC 3621
X-OITF-Content-Type	IETF RFC 3265, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-Content-Length	IETF RFC 3261

**Table 91 –HTTP extension headers for
the response to a SUBSCRIBE to presence (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261
X-OITF-Contact	IETF RFC 3621

Table 92 –HTTP extension headers for a SIP NOTIFY (IG→OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP SUBSCRIBE	IETF RFC 3621 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3621
X-OITF-To	IETF RFC 3621
X-OITF-Event	IETF RFC 3265, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-Call-ID	IETF RFC 3621
X-OITF-Subscription-State	IETF RFC 3265, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-CSeq	IETF RFC 3621
X-OITF-Content-Type	IETF RFC 3265 and OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-Content-Length	IETF RFC 3261
X-OITF-Contact	IETF RFC 3621

Table 93 – List of HTTP extension headers for a response to a received SIP NOTIFY (OITF→IG)

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3621 SIP/2.0 <response>
X-OITF-From	IETF RFC 3621
X-OITF-To	IETF RFC 3621
X-OITF-Call-ID	IETF RFC 3621
X-OITF-CSeq	IETF RFC 3621
X-OITF-Contact	IETF RFC 3621
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

NOTE Cancellation of subscription is not required if the X-OITF-Expires header was set to 0 in the initial SUBSCRIBE request.

6.5.4.3 Refreshing the subscription to the presence event – HTTP option

It is the responsibility of the application (in the OITF) initiating the subscription procedure to refresh the subscription according to the refresh subscription timer received in the response during the subscription process. Refreshing should be performed before the expiry of the refresh timer. A subscription that is not refreshed before the expiration of the refresh subscription timer shall be terminated by the network.

The procedure for refreshing the subscription to the presence event is the same as the procedure for subscribing to Presence.

6.5.4.4 Procedure for publishing presence information – HTTP option

This procedure for publishing an event may be invoked from either a DAE application or an embedded application in the OITF. The procedure is as follows:

- Step 1:** The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.
- HTTP request header including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 94.
- HTTP request body: as per 6.5.4.6.
- Step 2:** The IG shall validate that the request includes all the mandatory SIP headers for the publication process as per Table 94. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.
- Step 3:** The IG shall send a SIP PUBLISH to the network. When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the publish request. The response shall include a list of SIP headers as per Table 95, in addition to the normal HTTP headers as per IETF RFC 2616.

**Table 94 –HTTP extension headers
for the PUBLISH request (OITF→IG)**

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the IMS Public User Identity of the IPTV end user who has just registered.	IETF RFC 3261 PUBLISH <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line".	IETF RFC 3261
X-OITF-Event	IETF RFC 3261, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Expires	IETF RFC 3261, IETF RFC 3903
X-OITF-SIP-If-Match	IETF RFC 3903, IETF RFC 3903
X-OITF-Content-Type	IETF RFC 3261
X-OITF-Content-Length	IETF RFC 3261

**Table 95 –HTTP extension headers
for a response to SIP PUBLISH (IG→OITF)**

X-OITF HTTP header	Source of coding information
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-ETag	IETF RFC 3261, IETF RFC 3903 IETF RFC 3903
X-OITF-Expires	IETF RFC 3261

6.5.4.5 Procedure for refreshing published presence information – HTTP option

It is the responsibility of the OITF to refresh the published presence information before the refresh timer expires. A published event that is not refreshed shall be deleted in accordance with IETF RFC 3903.

6.5.4.6 Presence notification and publish schema

When the IPTV presence service is active, the body of the PUBLISH request shall include the extended OMA presence schema compliant to 5.1.6 of ETSI TS 183 063 “Procedure for IPTV presence service”.

In the extended OMA presence XML document, each service is described by the “service-description” OMA parameter as specified in OMA-ERP-Presence_SIMPLE-V1_1-20080627-A OMA-ERP-Presence_SIMPLE-V1_1-20080627-A. New “service-id” values are defined for IPTV with the following values:

- IPTV-BC: Scheduled Content service, defined in ETSI TS 183 063, 5.1.6.
- IPTV-CoD: Content on Demand service, defined in ETSI TS 183 063, 5.1.6.
- IPTV-NPVR: Network PVR service, defined in ETSI TS 183 063, 5.1.6.
- Broadcast-TV: DVB-T/H/C/S service, defined in Annex I.

When the user has an active IPTV presence service, the <tuple> element pertaining to the active service shall contain the corresponding element as defined in the presence schema.

6.5.5 Content sharing

6.5.5.1 Procedure for OITF content sharing capability query

Capability of the other terminal can be queried, i.e. the recipient is capable of supporting Content Share session or not. To perform a capability query, the OITF may use the following procedure:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 96

HTTP request body: empty

Step 2: The IG shall validate that the request includes all the mandatory HNI-IGI headers for the process as per Table 96. The IG shall reject a request that is missing any

mandatory HNI-IGI headers with a non-200 OK HTTP response, including the reason for rejection. The IG shall generate the SIP OPTIONS by mapping the X-OITF headers to the appropriate SIP header.

Step 3: The IG shall send a SIP OPTIONS to the network. The receiving OITF generates the response based on the black white lists. When the sender is in the black list (that means the sender has no right to query the receiving OITF's capability), the receiving OITF shall reject the request with a non-200 OK response, including the reason for rejection. When the sender is in white list (that means the sender has a right to query the receiving OITF's capability), the receiving OITF shall respond to the request with a 200 OK, including the information set of the receiving OITF which the sender has right to query.

Step 4: When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the SIP OPTIONS. The content of the HTTP response shall be:

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 96

HTTP request body: application/sdp

The response body includes the SDP offer generated by the OITF. SDP shall be used as specified in ETSI TS 124 503.

- An m=<media> <port> <transport> <fmt>
- An a=rtptime:<payload type> <encoding name>

The 200 OK body includes the SDP generated by the OITF which is the recipient of the OPTIONS to indicate supports for Content Share service. SDP shall be used as specified in ETSI TS 124 503.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

**Table 96 –HTTP extension headers on the HNI-IGI interface (OITF→IG)
for a SIP OPTIONS request and response (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request URI shall be set to the Public identity of the IPTV target end user	IETF RFC 3261 OPTIONS <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line OPTIONS" header	IETF RFC 3261
X-OITF-Contact Remarks: a) URI parameter shall be included and shall match what is sent in the Contact header included in the registration request. b) Expires parameter should be included.	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF- Accept Accept: application/sdp	IETF RFC 3261
X-OITF-Accept-Contact shall be set to +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel " and +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.gsma-vs".	[PRD IR.84]
X-OITF-Content-Length Set to 0	IETF RFC 3261

6.5.5.2 Invitation procedure for OITF originating a content share

To initiate a content sharing session to another OITF, the OITF shall use the following procedure:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 97

HTTP request body: application/sdp

The request body includes the SDP offer generated by the OITF. SDP shall be used as specified in ETSI TS 124 503.

- An m = <media> <port> <transport> <fmt>
- A c = <network type> <address type> <connection address> to indicate connection data at media level
- One or more a = fntp lines representing RTSP specific attributes
- An "a=" line with a "sendonly"
- An "a=X" -type: videolive/ videoclip, to be used to indicate the receiving user whether the Video is a live feed from the camera or whether the video is from a recorded clip.
- One or more a = file-selector lines which parameterize the file to be transferred

- An "b=" to indicate the bandwidth for each media stream

Step 2: The IG shall validate that the request includes all the mandatory HNI-IGI headers for the process as per Table 97. The IG shall reject a request that is missing any mandatory HNI-IGI headers with a non-200 OK HTTP response, including the reason for rejection. The IG shall generate the SIP INVITE by mapping the X-OITF headers to the appropriate SIP header.

Step 3: The IG shall send a SIP INVITE to the network. When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the SIP INVITE. The response includes a list of SIP headers as per Table 98 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 4: Upon receipt of a 200 OK response, the OITF shall send an HTTP PENDING_IG to acknowledge the final response. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 99

HTTP request body: empty

**Table 97 –HTTP extension headers on the HNI-IGI interface
(OITF→IG) for a SIP INVITE request**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request URI shall be set to the IMPU of the subscriber with whom the session is requested.	IETF RFC 3261 INVITE <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line INVITE" header	IETF RFC 3261
X-OITF-Contact Remarks: a) URI parameter shall be included and shall match what is sent in the Contact header included in the registration request. b) Expires parameter should be included.	IETF RFC 3261
X-OITF-Accept-Contact shall be set to +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel " and +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.gsma-vs".	PRD IR.84
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

Table 98 –HTTP extension headers for a 200 OK response received for the INVITE IG→OITF

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 <response>
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact	IETF RFC 3261

Table 99 –HTTP extension headers in HNI-IGI ACK request

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message.	IETF RFC 3261 ACK <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Contact The URI parameter shall be included, and shall match what has been inserted in the INVITE message. IG includes all other mandatory parameters that are absent.	IETF RFC 3261

6.5.5.3 Procedure for OITF refresh of a content sharing session

It is the responsibility of the OITF application to refresh the content sharing session before the session expires. The IG shall consider a session terminated if it is not refreshed.

6.5.5.4 Procedure for modification of a content sharing session

To perform session modification during an ongoing content sharing session, the OITF shall generate a re-INVITE request as defined in Table 97. The OITF shall include a new SDP offer in the session modification request reflecting the purpose of the session modification request.

Examples of events that can lead to session modification include bandwidth changes, putting the ongoing session on hold, sharing of a new content by any peer, etc.

6.5.5.5 Transferring a content sharing session

6.5.5.5.1 OITF target discovery

An OITF that wants to locate a target OITF for session transfer purposes shall perform the procedures described in 6.4.6.1.4.

Subsequently, a target OITF can be selected from the returned information.

6.5.5.5.2 Transferor OITF initiating a session transfer request – Push mode

The following procedure shall be supported for the receiving OITF to select the appropriate OITF for consumption of the content.

A transferor shall initiate the request for a content sharing session to transfer to an appropriate OITF. Upon successful transfer of the session, the content sharing service will be streamed to the target device.

To initiate a session transfer request, the transferor OITF shall follow the following procedure:

Step 1: As a pre-requisite it is assumed that the transferor OITF user is receiving a content sharing session and has selected a target device (transferee OITF) for the session.

In this step the transferor OITF shall include the transferee identifier, IMS service identifier and SDP in the SIP REFER (as shown in step 2) message.

Step 2: The transferor OITF shall send an HTTP POST request for the session transfer to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows:

HTTP request header including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 100 lists details of HTTP extension headers for an outgoing SIP REFER from the transferor for initiating up a session transfer request (OITF→IG) and incoming SIP REFER request to the transferee (IG→OITF).
- Table 100

HTTP request body: empty

Step 3: The IG shall validate that the request includes all the mandatory SIP headers as per Table 100 lists details of HTTP extension headers for an outgoing SIP REFER from the transferor for initiating up a session transfer request (OITF→IG) and incoming SIP REFER request to the transferee (IG→OITF).

Table 100 and send SIP REFER to the network. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection.

Step 4: The IG shall send a SIP REFER to the network, to setup the request and shall wait for the response. At some point in time, the IG shall return a HTTP 202. Accept response (or other appropriate responses) to the transferor OITF to report the received response from the transferee to the transfer request. The response shall include a list of SIP headers as per Table 47 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 5: Following that, the transferor OITF shall send an HTTP HNI-IGI PENDING_IG request (refer to 6.6.1), and shall wait for any response reporting the outcome of the session transfer procedure.

Step 6: At some point in time, the IG shall receive an incoming SIP NOTIFY from the remote OITF, reporting the outcome of the session transfer and which it shall forward to the transferor OITF in an HTTP 200 OK response. The HTTP response shall include the list of SIP headers as per Table 101 in addition to the normal HTTP headers. The body of the HTTP response shall include the SDP body received in the NOTIFY.

Step 7: The transferor OITF shall return the SIP 200 OK response, acknowledging the SIP NOTIFY, to the IG, in an HTTP POST PENDING_IG request. The content of the HTTP request shall be as follows.

HTTP request header including the following:

- <list of HTTP headers> – as per IETF RFC 2616

- <list of SIP headers encoded as HTTP headers> – as per Table 49

HTTP request body: empty

Table 100 lists details of HTTP extension headers for an outgoing SIP REFER from the transferor for initiating up a session transfer request (OITF→IG) and incoming SIP REFER request to the transferee (IG→OITF).

Table 100 –HTTP extension headers for SIP REFER

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall be set to the identifier of the transferee (target OITF).	IETF RFC 3261 REFER <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To The URI part of X-OITF-To shall be set to the value of the request URI in the "X-OITF-request-Line".	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Refer-To shall be set to the remote target URI included in the contact header field returned in the SIP 200 OK associated with initial session setup with the transferor and extended with the following URI header fields: Replaces header field shall include the SIP dialog identifier for the original content sharing session as per IETF RFC 3891. Require header field populated with the option tag value "replaces". To header field shall contain the original content identifier copied from the request URI of the original SIP INVITE request initiated from the transferor. An Accept-Contact header field shall be set to +g.3gpp.icsi-ref="urn%3Aurn-7%3A-3gpp-service.ims.icsi.mmtel" and +g.3gpp.iari-ref="urn%3Aurn-7%3A-3gpp-application:ims.iari.gsma-vs". Body header. Contains the SDP body to be included in the SIP request initiated from the transferor OITF. The SDP body shall contain the same number of media lines as the SDP used in the original session from the transferor OITF. Each media line shall indicate the same media type as its corresponding media component in the SDP used in the original session by the transferor OITF. The media line for the media to be transferred shall include a port number with non zero value.	IETF RFC 3261 IETF RFC 3515 IETF RFC 3891
X-OITF-Contact	IETF RFC 3261

Table 101 – Supported HTTP extension headers in the NOTIFY request sent from the transferee OITF (OITF->IG) to the transferor (IG->OITF)

X-OITF HTTP header	Source of coding information
X-OITF-request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP REFER	IETF RFC 3261 NOTIFY <request URI> SIP/2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Event	IETF RFC 3515
X-OITF-Call-ID	IETF RFC 3261
X-OITF-Subscription-State	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Length	IETF RFC 3261

6.5.5.5.3 Remote OITF receiving an incoming session transfer request – Push mode

The remote OITF is having a content sharing session with transferor OITF. The procedure for remote OITF in a push mode is as follows:

Step 1: It is assumed that the remote OITF has an HTTP PENDING_IG request. At some point in time, when a REFER request targeted for the transferee OITF is received by the IG, the IG shall return a HTTP 200 OK response to the OITF. The response shall include the list of SIP headers as per Table 100 lists details of HTTP extension headers for an outgoing SIP REFER from the transferor for initiating up a session transfer request (OITF->IG) and incoming SIP REFER request to the transferee (IG->OITF).

Table 100 in addition to the normal HTTP headers as per IETF RFC 2616. The body of the HTTP response shall include the XML structure as per 6.3.13.2.

Step 2: The remote OITF shall examine the incoming REFER request including the security checking and IMS Content Sharing service identifying. In particular, the OITF shall extract the body header to use it to later construct its own SDP for the session transfer. If the OITF cannot successfully validate the extracted SDP, it shall reject the incoming request. If the OITF successfully validates the extracted SDP, it should accept the incoming request.

Step 3: Once the remote OITF accepts the incoming SIP REFER, it shall send an HTTP POST PENDING_IG request to the IG. The content of the HTTP request shall be as follows.

HTTP request header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 47 with the exception that the response in this case is an SIP 202 OK

HTTP request body: empty

Step 4: The remote OITF shall extract the following information from the incoming REFER request:

- the Content URI extracted from the To header included in the Refer-To header;
- the body header;
- the Dialog ID to be replaced extracted from the Replace header in the Refer-To header;
- the m-line extracted from SDP, m-line indicating the multimedia session information of content sharing.

- Step 5:** At some point in time, the remote OITF shall then construct an SDP that it can use to initiate a new session to the transferee OITF. The remote OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6.
- Step 6:** The IG shall validate that the request includes all the mandatory HNI-IGI headers. The IG shall reject a request that is missing any mandatory HNI-IGI headers with a non-200 OK HTTP response, including the reason for rejection. The IG shall generate the SIP INVITE by mapping the X-OITF headers to the appropriate SIP header as per Table 97, requesting to set up a new session with transferee OITF.
- Step 7:** The IG shall send a SIP INVITE to the network. When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the OITF to report the response to the SIP INVITE. The response includes a list of SIP headers as per Table 98 in addition to the normal HTTP headers as per IETF RFC 2616.
- Step 8:** Upon receipt of a 200 OK response, the remote OITF shall send an HTTP PENDING_IG to acknowledge the final response. The content of the HTTP request shall be as follows.
- HTTP request header: including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 99
- HTTP request body: empty
- Step 9:** At some point in time, the IG shall receive an incoming SIP NOTIFY from the remote OITF, reporting the outcome of the session transfer and which it shall forward to the transferor OITF in an HTTP 200 OK response. The HTTP response shall include the list of SIP headers as per Table 101 in addition to the normal HTTP headers. The body of the HTTP response shall include the SDP body received in the NOTIFY.
- Step 10:** At some point in time, IG shall receiving the SIP 200 OK response from the transferor OITF indicating the acknowledgement of the SIP NOTIFY, in an HTTP POST PENDING_IG request. The content of the HTTP request shall be as follows.
- HTTP request header including the following:
- <list of HTTP headers> – as per IETF RFC 2616
 - <list of SIP headers encoded as HTTP headers> – as per Table 49
- HTTP request body: empty

6.5.5.5.4 Transferee OITF receiving an incoming session transfer request – push mode

The remote OITF has a content sharing session with transferor OITF. The procedure at the transferee OITF receiving an incoming session transfer request in a push mode is as follows:

- Step 1:** It is assumed that the transferee OITF has an HTTP PENDING_IG request. At some point in time, when an INVITE request targeted for the transferee OITF is received by the IG, the IG shall return a HTTP 200 OK response to the transferee OITF. The response shall include the list of SIP headers as per Table 97 in addition to the normal HTTP headers as per IETF RFC 2616.
- Step 2:** The transferee OITF shall examine the incoming INVITE request including the security checking and IMS Content Sharing service identifying. In particular, the transferee OITF shall extract the body header for the session transfer. If the OITF cannot successfully validate the extracted SDP, it shall reject the incoming request. If the OITF successfully validates the extracted SDP, it should accept the incoming request.
- Step 3:** When the IG receives the response, the IG shall return a HTTP 200 OK response (or other appropriate responses) to the remote OITF to report the response to the SIP INVITE. The response includes a list of SIP headers as per Table 98 in addition to the normal HTTP headers as per IETF RFC 2616.

Step 4: Upon receipt of a 200 OK response, the remote OITF shall send an HTTP PENDING_IG to acknowledge the final response. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 99

HTTP request body: empty

6.5.5.6 Procedure for OITF terminating a content sharing session

In order to terminate a content sharing session, the OITF shall use the following procedure:

Step 1: The OITF shall send an HTTP POST request to the IG over the HNI-IGI interface, as described in 6.6. The content of the HTTP request shall be as follows.

HTTP request header: including the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <list of SIP headers encoded as HTTP headers> – as per Table 102

HTTP request body: empty

Step 2: The IG shall validate that the request includes all the mandatory SIP headers for the message as per Table 102. The IG shall reject a request that is missing any mandatory SIP headers with a non-200 OK HTTP response, including the reason for rejection. The IG shall generate the SIP BYE by mapping the X-OITF headers to the appropriate SIP headers.

Step 3: The IG shall send a HTTP 200 OK response to the OITF when the SIP 200 OK is received as a response to the content sharing session termination request. The SIP 200 OK headers are mapped as indicated in Table 103 in addition to the normal HTTP 200 OK headers.

**Table 102 – HTTP extension headers
for a SIP BYE request (OITF→IG)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-request-Line Note that the request URI shall be set to the contact return in the 200 OK for the invite.	IETF RFC 3261 BYE <request URI> SIP/ 2.0
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Contact shall be set to the value received in the contact of a 200 OK for session termination or SIP INVITE for session origination	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261
X-OITF-Content-Length Shall be set to 0	

**Table 103 –HTTP extension headers for
a 200 OK response to a BYE (IG→OITF)**

X-OITF HTTP header	Source of information for coding purposes
X-OITF-response-Line	IETF RFC 3261 SIP/2.0 200 OK
X-OITF-From	IETF RFC 3261
X-OITF-To	IETF RFC 3261
X-OITF-Call-ID	IETF RFC 3261
X-OITF-CSeq	IETF RFC 3261

6.6 Protocols system infrastructure functions OITF-IG interface (HNI-IGI)

6.6.1 HNI-IGI message types

The HTTP protocol is used to exchange information between the IG and the OITF. The IG behaves as an HTTP server and the OITF behaves as an HTTP client. The OITF part may be implemented either in native code or in DAE applications.

There are several aspects of information on the HNI-IGI interface.

- Normal HTTP headers.
- Application-specific information that is translated by the IG into SIP headers. These are included as HTTP extension headers and have the same name as in the SIP message, but are prefixed with X-OITF.
- Application-specific information that forms the body of a SIP message. This corresponds to the SIP message body and is included as a body in the HTTP request or response. An example message body type is SDP.
- HNI-IGI auxiliary information that is only used between OITF and IG. These parameters are prefixed with X-HNI-IGI. An example is those parameters related to the fetching of GBA credentials by the OITF for re-use of GBA authentication mechanism for single sign-on (see 6.4.6.2.3).

The general format of an HNI-IGI HTTP request is

```
HTTP POST <IG URI>/<HNI-IGI message type>
<HTTP headers>
<X-OITF extension headers> or <X-HNI-IGI extension headers>
Content-Type: <...>
Content-Length: <Number>
<message body>
```

The general format of an HNI-IGI HTTP response is

```
HTTP/1.1 <HTTP response>
<HTTP headers>
<X-OITF extension headers> or <X-HNI-IGI extension headers>
Content-Type: <...>
Content-Length: <Number>
<message body>
```

Table 104 lists the HNI-IGI message types.

Table 104 – HNI-IGI message types

HNI-IGI message type	Meaning
PENDING_IG	The message is a pending HTTP request, that shall only be responded to by the IG when it needs to contact the OITF as a result of an incoming request from the network (e.g. an incoming MESSAGE).
SIP	The message is an HNI-IGI message corresponding to a SIP message. The IG shall translate this into a corresponding SIP message by adding and changing the relevant headers.
AUX	The message is an HNI-IGI message that does not translate to a SIP message. The IG processes this message and responds accordingly.

Messages over the HNI-IGI interface can be sent in both directions.

- Normal HTTP requests are used for requests from the OITF and responses from the IG.
- There shall be an HTTP request from the OITF to the IG with the response pending to allow new (unsolicited) messages from the network to be sent from the IG to the OITF in the response. This is a special kind of HNI-IGI message, called PENDING_IG.

An example of SIP header that is mapped to an HTTP extension header is: "From: david@oiptv.org" that becomes "X-OITF-From:david@oiptv.org".

Note that only AUX HNI-IGI message types are used by the HNI-IGI SIP option (see 6.6.6).

6.6.2 HNI-IGI messages in the OITF to IG direction – HTTP option

When the IG receives an HNI-IGI message, it shall add or change all SIP headers that are not specific to the application (Tags, Call ID, via, request URI etc.) while translating from HTTP to SIP.

Table 105 lists header values for the HNI-IGI protocol that the IG and OITF shall support and lists the IG action on those headers.

**Table 105 – X-OITF HTTP extension headers
and IG actions for OITF→IG messages**

HNI-IGI header	Description	IG action
X-OITF-request-Line	This is a special header which contains the SIP method and request URI for the corresponding SIP message, when the SIP message is a request, e.g. X-OITF-request-Line PUBLISH sip:david@oipvtv.org SIP/2.0.	The IG shall map this field to the SIP request line.
X-OITF-response-Line	This is a special header which contains the response line of the corresponding SIP message, when the SIP message is a response, e.g. 200 OK.	The IG shall map this field to construct the SIP response line.
X-OITF-Call-ID	Keeps track of sessions and dialogs.	The IG shall use this field internally between an OITF and the IG to keep track of sessions. The IG replaces it with a value maintained in SIP state machine on the SIP side.
X-OITF-Contact	Each method has its own use of the Contact field.	The IG shall map to the corresponding SIP header. The IG may add other parameters.
X-OITF-CSeq	Used to keep track of requests and responses.	IG shall use this field internally between an OITF and the IG to keep track of requests and responses, and replace it with a value maintained by the IG on the SIP side. The IG shall include the same value in subsequent responses to the OITF. The OITF shall respond with an error code if the value is incorrect.
X-OITF-From		The IG shall map to the corresponding SIP header. The IG may add information in sub-fields.
X-OITF-Event		The IG shall map to the corresponding SIP header.
X-OITF-Expires		The IG shall map to the corresponding SIP header.
X-OITF-To		The IG shall map to the corresponding SIP header.
X-OITF-Content-Type		The IG shall map to the corresponding SIP header, and shall match it with the actual body included in the HTTP request.
X-OITF-Content-Length		The IG shall verify the length of the message and insert the value in the SIP message.
X-HNI-IGI-request	This header specifies the request type of the HNI-IGI message.	See appropriate subclauses.

The above headers are not present in all HNI-IGI messages, and are not the only headers that can be present.

The OITF shall use an IMPU in X-OITF headers where an IMPU is required in the SIP header.

The OITF may include other headers that are application specific (e.g. X-OITF-Accept-Contact) in which case the IG shall include them transparently in the SIP method as long as they comply with the appropriate syntax for the header. Reference should be made to the various services using the HNI-IGI interface for a list of the headers that shall be present.

6.6.3 HNI-IGI messages in the IG to OITF direction – HTTP option

When the IG translates a SIP message to an HNI-IGI HTTP message, it shall remove SIP headers that should not be transmitted on the HNI-IGI interface, while translating from SIP to HTTP.

Table 106 lists header values in the SIP protocol that the IG and OITF shall support and the action the IG undertakes when mapping to the HNI-IGI protocol.

Table 106 – Mapping of SIP header to X-OITF HTTP extension headers in IG→OITF

SIP header	Description	IG action
Request Line (first line of SIP request message)	The request Line contains the method and a SIP URI.	The IG shall use this field to construct the X-OITF-request-Line
Response Line (first line of SIP response message)	The response Line contains the response code and SIP version information.	The IG shall use this field to construct the X-OITF-response-Line
Call-ID	Keeps track of sessions and dialogs.	The IG shall replace this with value used between IG and OITF in the X-OITF-Call-ID.
Contact		The IG shall map to the corresponding HNI-IGI header.
CSeq		The IG shall use this field to keep track of requests and responses on the HNI interface. The IG shall replace it with a value maintained in the IG. The IG shall include the same value in subsequent responses to the OITF. The OITF shall respond with an error code if the value is smaller than the previous one.
From		The IG shall map to corresponding HNI-IGI header. The IG may add information in sub-fields.
Event		The IG shall map to the corresponding HNI-IGI header.
Expires		The IG shall map to the corresponding HNI-IGI header.
To		The IG shall map to the corresponding HNI-IGI header.
Content-Type		The IG shall map to the corresponding HNI-IGI header.
Content-Length		The IG shall verify the length of the message and insert value in the HNI-IGI message.

The above headers may not be present in all HNI-IGI messages.

The IG shall map any other received SIP headers by adding X-OITF- to the specific SIP header. Reference should be made to the various services using the HNI-IGI interface for a list of the headers that shall be present.

The IG handles the SIP state machines.

6.6.4 HNI-IGI PENDING_IG message – HTTP option

6.6.4.1 General

HNI-IGI PENDING_IG messages are sent by DAE and embedded applications in the OITF whenever these applications are ready to receive any incoming message from the network.

PENDING_IG messages may include a SIP request or a SIP response. In this case, there is typically an ongoing SIP dialog between the OITF application and a peer SIP end-point in the IMS network.

HTTP headers included in a PENDING_IG message that includes a SIP request or a SIP response are the <X-OITF extension headers> that are pertinent to the application. The content of such a message shall be as follows:

HTTP request header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-OITF extension headers> Any number of those extension headers depending on the application

HTTP request body: <SIP message body if applicable>

PENDING_IG messages that do not include an SIP request or an SIP response are typically sent by applications in the OITF that do not have any ongoing communication with an SIP peer but are prepared to handle incoming requests for the IPTV user associated with any active application running in the OITF, or applications that have an ongoing SIP dialog with a SIP peer and are prepared to receive any SIP messages within that dialog. The content of PENDING_IG messages that do not include an SIP request or an SIP response shall be as follows.

HTTP request header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-OITF-Call-ID> Set to NULL for applications without an ongoing dialog or set to the proper value for applications with an ongoing SIP dialog.
- <X-OITF-From> Set to the IMPU of the target user associated with any active application in the OITF for applications without an ongoing dialog. Not needed for applications with an ongoing SIP dialog.

HTTP request body: empty

Note that once the target OITF application accepts an incoming request, the Call-ID in the outgoing response shall be set to the value used by the IG in its request to the OITF.

The content of the HTTP response to either of the above requests shall be as follows:

HTTP response header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-OITF extension headers> Any number of those extension headers depending on the application

HTTP response body: <Appropriate message body if applicable>

6.6.4.2 Refreshing of HNI-IGI PENDING_IG message

HNI-IGI PENDING_IG messages shall have to be refreshed periodically by the OITF. The refresh time shall be maintained in the IG and shall not exceed a SIP session Expiry timer, or a SIP subscription Refresh timer for the SIP session under consideration.

To enable an OITF to refresh an HNI-IGI PENDING_IG request, the IG shall, upon timer expiry, send an HTTP 200 OK response that does not include any X-OITF-<extension headers>.

Upon receipt of such a response, the OITF may decide to resend a new HNI-IGI PENDING_IG request or simply gracefully terminate the session.

6.6.4.3 Cancelling an HNI-IGI PENDING_IG message

The IG considers an HNI-IGI PENDING_IG request cancelled should it encounter one of the following events:

- the TCP connection on which the HNI-IGI PENDING_IG request was received is explicitly disconnected or timed out;
- the IG receives an HNI IGI SIP request to terminate the session (a SIP BYE) from the OITF application with the outstanding HNI-IGI PENDING_IG request;
- the IG receives an HNI IGI SIP request to terminate an ongoing subscription (a SIP SUBSCRIBE with X-OITF-Expiry set to 0) from the OITF application with the outstanding HNI-IGI PENDING_IG request.

For the last two cases, the IG shall send an HTTP 200 OK response that does not include any X-OITF-<extension-headers> as a response to the PENDING_IG request. Optionally, the IG may empty its internal buffers that may include in-transit messages destined for the applications.

6.6.5 HNI-IGI SIP message – HTTP option

HNI-IGI SIP messages are sent by DAE and embedded applications in the OITF whenever these applications are ready to send a SIP request or a SIP response. The content of such a message shall be as follows.

HTTP request header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-OITF extension headers> Any number of those extension headers depending on the application

HTTP request body: <SIP message body if applicable>

The content of the HTTP response shall be as follows:

HTTP response header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-OITF extension headers> Any number of those extension headers depending on the application

HTTP response body: <SIP message body if applicable>

6.6.6 HNI-IGI auxiliary message

HNI-IGI auxiliary messages are sent by DAE and embedded applications in the OITF whenever these applications are ready to send messages that are neither SIP type nor PENDING_IG type (for example fetching GBA credentials).

The content of such a message shall be as follows:

HTTP request header: it includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI-request> – Identifies the request

- <X-HNI-IGI extension headers> Any number of those extension headers depending on the request

HTTP request body: <application message body if applicable>

The content of the HTTP response shall be as follows:

HTTP response header: It includes the following:

- <list of HTTP headers> – as per IETF RFC 2616
- <X-HNI-IGI extension headers> – Any number of those extension headers depending on the request

HTTP response body: <application message body if applicable>

Note that only one auxiliary message, GBA-Registration, applies to both the HTTP and SIP options. All other auxiliary messages apply only to HTTP option.

6.6.7 HNI-IGI message body – HTTP option

The HNI-IGI messages in either direction shall include transparently the appropriate SIP body for the different SIP methods in the HTTP message body.

6.6.8 Guidelines for applications using the HNI-IGI interface – HTTP option

This subclause lists some guidelines that apply to DAE application and OITF applications that use the HNI-IGI interface. Both types of application are referred to as application here:

- It is the responsibility of the application to ensure that it provides all the SIP headers that are required for the correct operation of the application.
- The IG shall absorb “100 Trying” responses received from the network and not return them to the OITF.
- The IG shall transparently handle X-OITF-SIP headers received over the HNI-IGI interface unless specifically stated in this standard.
- It is the responsibility of the application to generate an ACK as the 2XX final response of an INVITE transaction; for non-2XX responses, the IG shall generate the ACK.
- Validation (of message structure and XML schema) of received XML data shall be the responsibility of the application.
- The application should ensure that a SIP method is supported by the IG before using it.
- The IG shall return a 405 Method Not Allowed error fault if it receives a request over the HNI-IGI that includes a SIP method that it does not support.
- The Pending request shall be refreshed as per 6.6.4.2.
- An HTTP pending request sent to the IG may include a SIP response (or SIP request if no response is expected) to be transmitted to the SIP network.
- The only identified case where a PENDING-IG request shall include an SIP request is the case where the OITF application sends an ACK.

6.6.9 Error recovery in the IG – HTTP option

This subclause covers the handling in the IG for encountered error cases.

If the IG detects a timeout, or an explicit disconnection on all TCP links between an OITF application and the IG, the IG shall consider the application inactive. The application may become active again by re-establishing the TCP link with the IG. Any incoming SIP messages destined for an application that is inactive shall result in an error message 487 “Request Terminated” (481 “Call Leg/Transaction Does Not Exist” is also an acceptable response if the IG concludes that the application is permanently inactive) being returned to the network as a

response. For an inactive application, the SIP dialog in the IG shall eventually time-out and clear all resources in the IG and the network.

In order to cater to the scenario of transient TCP link disconnects, and to allow incoming messages that are received by the IG during the time it takes the OITF to re-establish a TCP link following a disconnection, it is recommended that the IG waits no more than 1 s before it responds to the network with a 487 or a 481 error message.

In cases where the WANGW is integrated into the IG (herein referred to as IG-WANGW), the IG-WANGW shall detect that an OITF is restarted upon receipt of a DHCP server discovery request (DHCPDISCOVER message) and IP address request (DHCPREQUEST message), and where the IG-WANGW internal state indicates that the OITF is powered on. In such a case, the IG-WANGW shall terminate all active SIP sessions, the IG-WANGW shall de-register all users that are logged in from the restarted OITF as stored in the IG-WANGW state. Note that the IG-WANGW is able to keep a mapping between the SIP dialogs ongoing, the IMPUs of registered users and the IP addresses and deviceIDs of the devices being used. Following the deletion of stale SIP state and de-registration of users, the IG-WANGW shall act on the OITF start up high level procedure requests.

If the IG does not have access to the DHCP information exchanged between the OITF and the WANGW, refer to 6.4.6 in case the OITF undergoes a restart.

The OITF shall detect that an IG is restarted when all TCP links to the IG timeout simultaneously, or are explicitly disconnected. If this is the case, the OITF shall refrain from sending any message to the IG until such time that the OITF detects that the IG has restarted, using UPnP IG discovery procedure in that regard (polling). Following that, the OITF shall execute steps 2-6 of the "OITF Start up High-Level Procedure".

For a transition period following an IG restart, active SIP sessions in the network (that will eventually timeout and be cleared) that are no longer active in the IG, will continue to send SIP messages to the IG, to which the IG shall respond with error message 481, "Call leg/transaction does not exist".

6.7 Protocols for content preparation

6.7.1 Reference points NPI-45, NPI-46, NPI-CSPT3 and NPI-CSPG3

6.7.1.1 General

The reference points specified here are introduced in Annex B of IEC 62766-1:2017.

6.7.1.2 Overview

Subclause 6.7.1 describes the reference points NPI-45 and NPI-46 between CoD or Scheduled Content Encryption Functions and Key Management Function. Encryption Functions call this interface when managing keys and DRM signaling for CoD and unicast scheduled content. For the case of multicast scheduled content, the appropriate interface is the ECMG $\Leftrightarrow$ SCS interface defined in ETSI TS 103 197.

The interface described in this subclause is exposed by the Key Management Function which is in charge of:

- synchronizing content keys between DRM by exposing services allowing:
 - the Encryption Functions to retrieve the keys that have to be used to encrypt a content,
 - the Encryption Functions to provision keys (to Key Management Function), if keys are generated by the Encryption Functions or another entity;
- centralizing the signaling that has to be inserted in the media for each DRM.

These interfaces apply in particular when content is used in a multi-DRM mode where the content is encrypted once and the corresponding keys are shared by all the DRM. Signaling specific to every DRM may be inserted in the content.

This interface specification is also applicable to the reference points NPI-CSPT3 and NPI-CSPG3. In this case, the interface described in this subclause is exposed by the CSP-T Server or CSP-G Server, which are in charge of

- retrieving or providing keys used to encrypt a content,
- providing the signaling that has to be inserted in the media for a given DRM.

6.7.1.3 Interface implementation

6.7.1.3.1 General

The interface is implemented as a WEB service. The associated classification scheme is provided in 6.7.2.

Each reference point identifies a calling entity and a called entity as shown in Table 107.

Table 107 – Reference points calling entity and called entity

Reference point	Calling entity	Called entity
NPI-45	CoD Encryption Function	Key Management Function
NPI-46	Scheduled Content Encryption Function	Key Management Function
NPI-CSPT3	Key Management Function	CSP-T Server
NPI-CSPG3	Key Management Function	CSP-G Server

Three operations are defined. They are intended to be used according to the role of the functional entities as shown in Table 108.

Table 108 – Operations according to functional entity roles

	Key rotation handled by called entity	Key rotation handled by calling entity
Key generation handled by called entity	getKeyScheduleSignaling	setScheduleAndGetKeySignaling
Key generation handled by calling entity	This case is not supported.	setKeyScheduleAndGetSignaling

6.7.1.3.2 getKeyScheduleSignaling operation

6.7.1.3.2.1 Method and parameters

Table 109 shows the getKeyScheduleSignaling parameters.

Table 109 – getKeyScheduleSignaling method parameters

Operation name		Description
getKeyScheduleSignaling		Operation name
Field name	Field type	Description
GetKeyScheduleSignalingrequest	GetKeyScheduleSignalingrequestType	Input parameter
GetKeyScheduleSignalingresponse	GetKeyScheduleSignalingresponseType	Output parameter

6.7.1.3.2.2 Input parameter structure definition

The XML schema of the GetKeyScheduleSignalingrequest message is shown in Figure 5 and the GetKeyScheduleSignalingrequestType is shown in Table 110.

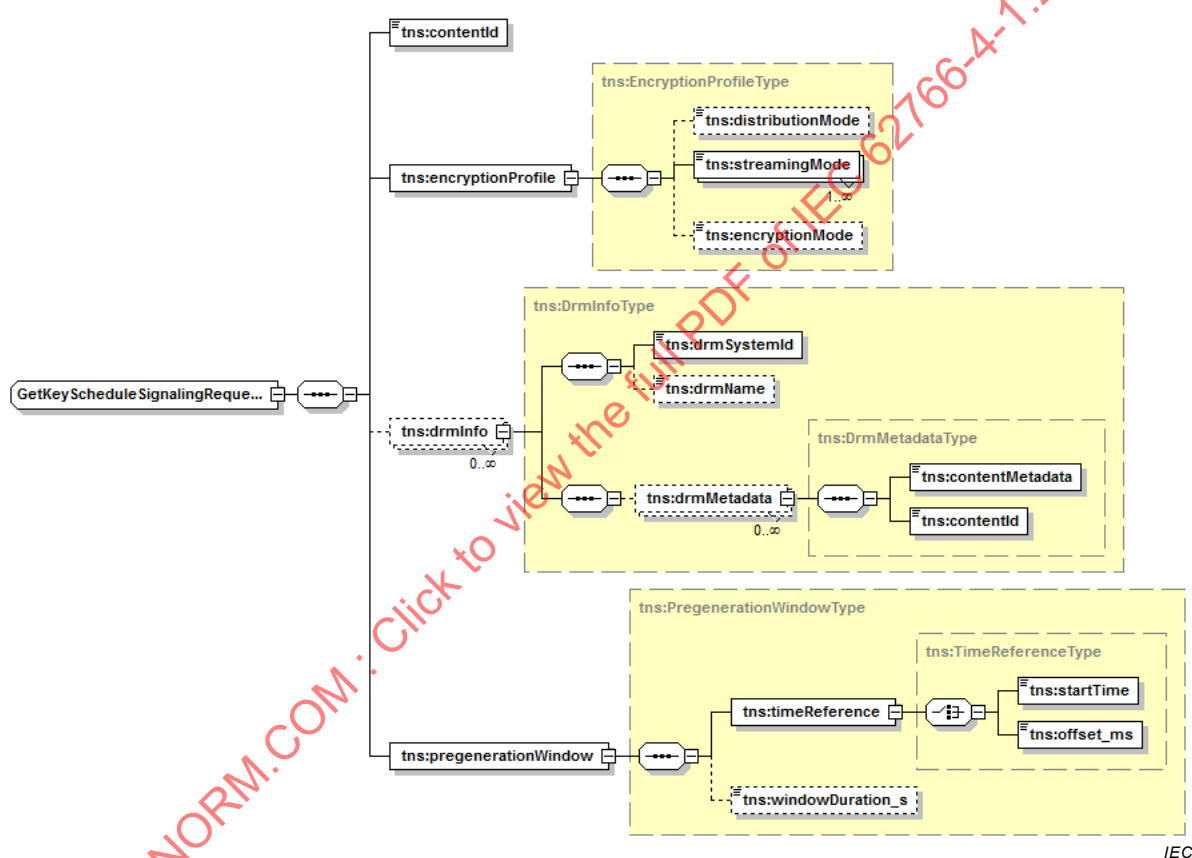
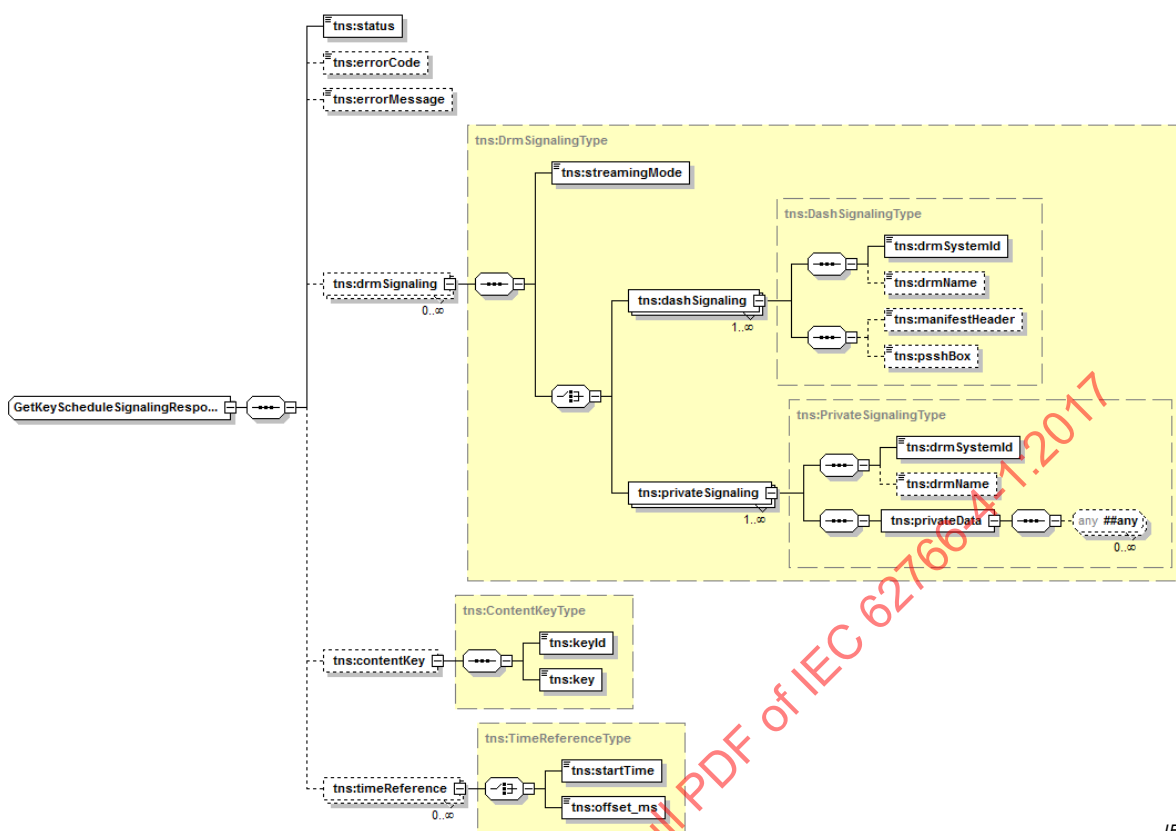
**Figure 5 – GetKeyScheduleSignalingrequest schema**

Table 110 – GetKeyScheduleSignalingrequestType

Field name	Field type	Multiplicity	Description	Reference
contentId	xs:string	1	Identifier for a content, a content being either a scheduled content service, an event on a scheduled content service (i.e. with access criteria specific to the event) or a CoD asset. It is provided by the Scheduling Function (for Scheduled content) or the Content Management Function (for CoD).	
encryptionprofile	EncryptionprofileType	1	Set of properties for the management of the content key and the signaling.	6.7.1.3.6.6
drmInfo	DrmInfoType	0..N	Information for addressed DRM systems. This allows the calling entity to provide a set of DRM systems for which the drmSignaling shall be provided. An empty list means that the called entity manages a DRM list from its own configuration.	6.7.1.3.6.2
pregenerationWindow	PregenerationWindowType	1	The time information for the Key Management Function to know the period it is expected to generate keys for.	6.7.1.3.6.8

6.7.1.3.2.3 Output parameter structure definition

Figure 6 shows the XML schema of the GetKeyScheduleSignalingresponse message and Table 111 shows the GetKeyScheduleSignalingresponseType.



IEC

Figure 6 – GetKeyScheduleSignalingresponse schema

Table 111 – GetKeyScheduleSignalingresponseType

Field name	Field type	Multiplicity	Description	Reference
status	StatusType	1	The global status of the request.	6.7.1.3.5.2
errorCode	xs:unsignedInt	0..1	The code of the error in case the status is not OK. Values are specific to the implementation of the called entity. It is omitted if the request succeeds.	
errorMessage	xs:string	0..1	Any additional information related to the status in case of errors. It is omitted if the request succeeds.	
timeReference	TimeReferenceType	0..N	The dates and times for which keys have been generated. It is omitted if the request failed.	6.7.1.3.6.12
contentKey	ContentKeyType	0..1	The content key to be used to protect the content. It is omitted if the request failed.	6.7.1.3.6.1
drmSignaling	DrmSignalingType	0..N	The signaling information according to every provided streaming mode. This information allows the CSP client to find the key materials required to decrypt a given content, either locally when the key has already been retrieved or remotely when requested for the first time. Each DRM will provide signaling information for a given [content/content key] pair, such as URL of the CSP server or DRM. The DRM information is DRM specific. It is omitted if the request failed.	6.7.1.3.6.4

The called entity generates the keys based on its schedule and provides:

- the key (i.e. contentKey element) to use at the requested time (i.e. timeReference element in the pregenerationWindow element of the request) provided by the calling entity;
- the signaling (i.e. list of drmSignaling elements) to be inserted by the Encryption Function when using the key;

- the times (i.e. timeReference element) for the current and next keys encompassing the window duration (i.e. windowDuration element in pregenerationWindow element of the request); these times should be used to perform the subsequent calls to this method.

One drmSignaling element is provided for every provided streamingMode. For each streamingMode, either a dashSignaling or a privateSignaling element is provided for every addressed DRM. This element holds the drmSystemId and the signaling information (defined by the encryptionprofile). For DASH and CENC, either a manifestheader (XML fragment) or a PSSH box data or both can be provided. The manifestheader is to be inserted in the MPD, the psshBox is to be inserted in the content.

If an error occurs, then no contentKey, no timeReference, and no drmSignaling elements will be provided.

6.7.1.3.3 setScheduleAndGetKeySignaling operation

6.7.1.3.3.1 Method and parameters

Table 112 shows the setScheduleAndGetKeySignaling method parameters.

Table 112 – setScheduleAndGetKeySignaling method parameters

Operation name		Description
setScheduleAndGetKeySignaling		Operation name
Field name	Field type	Description
SetScheduleAndGetKeySignalingrequest	SetScheduleAndGetKeySignalingrequestType	Input parameter
SetScheduleAndGetKeySignalingresponse	SetScheduleAndGetKeySignalingresponseType	Output parameter

6.7.1.3.3.2 Input parameter structure definition

Figure 7 is the XML schema of the SetScheduleAndGetKeySignalingrequest message and Table 113 shows the SetScheduleAndGetKeySignalingrequestType.

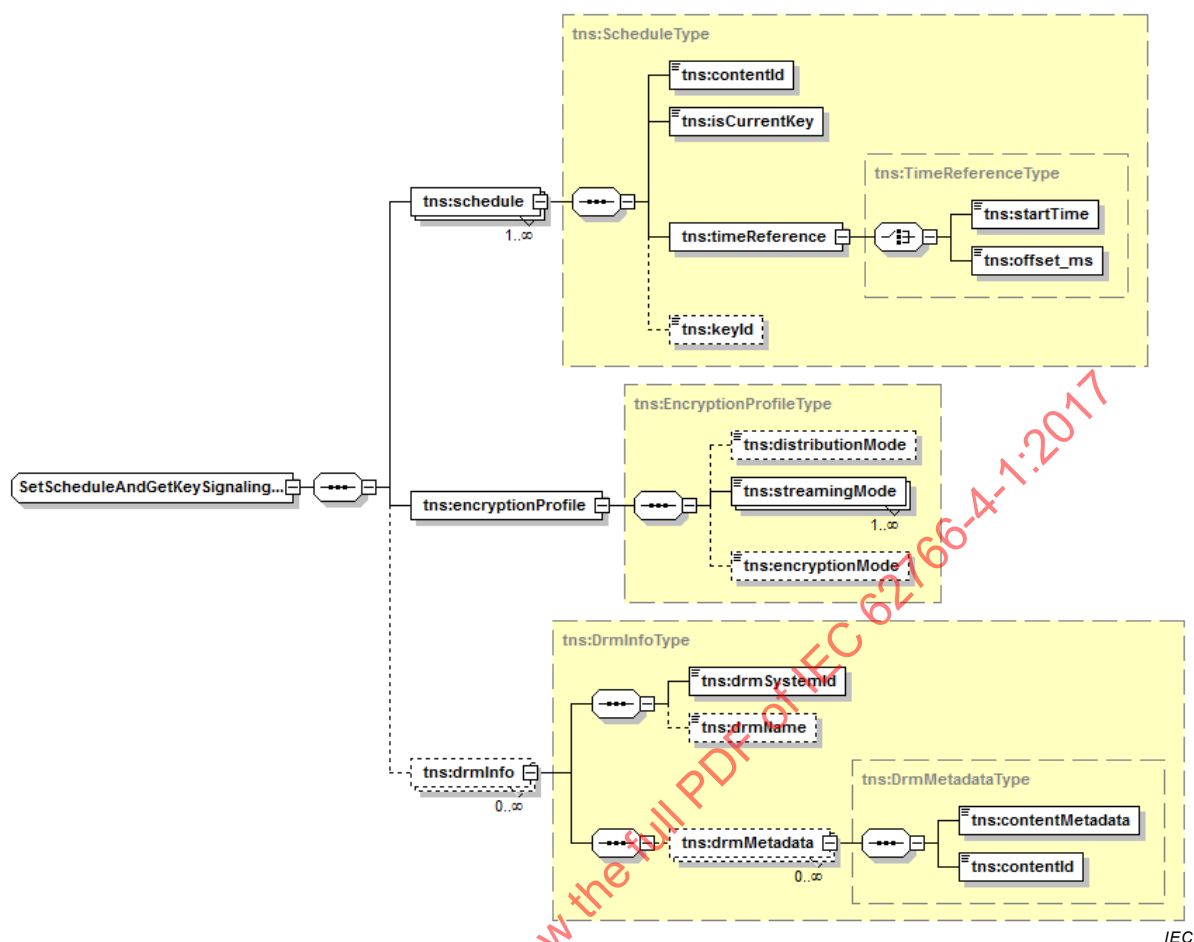


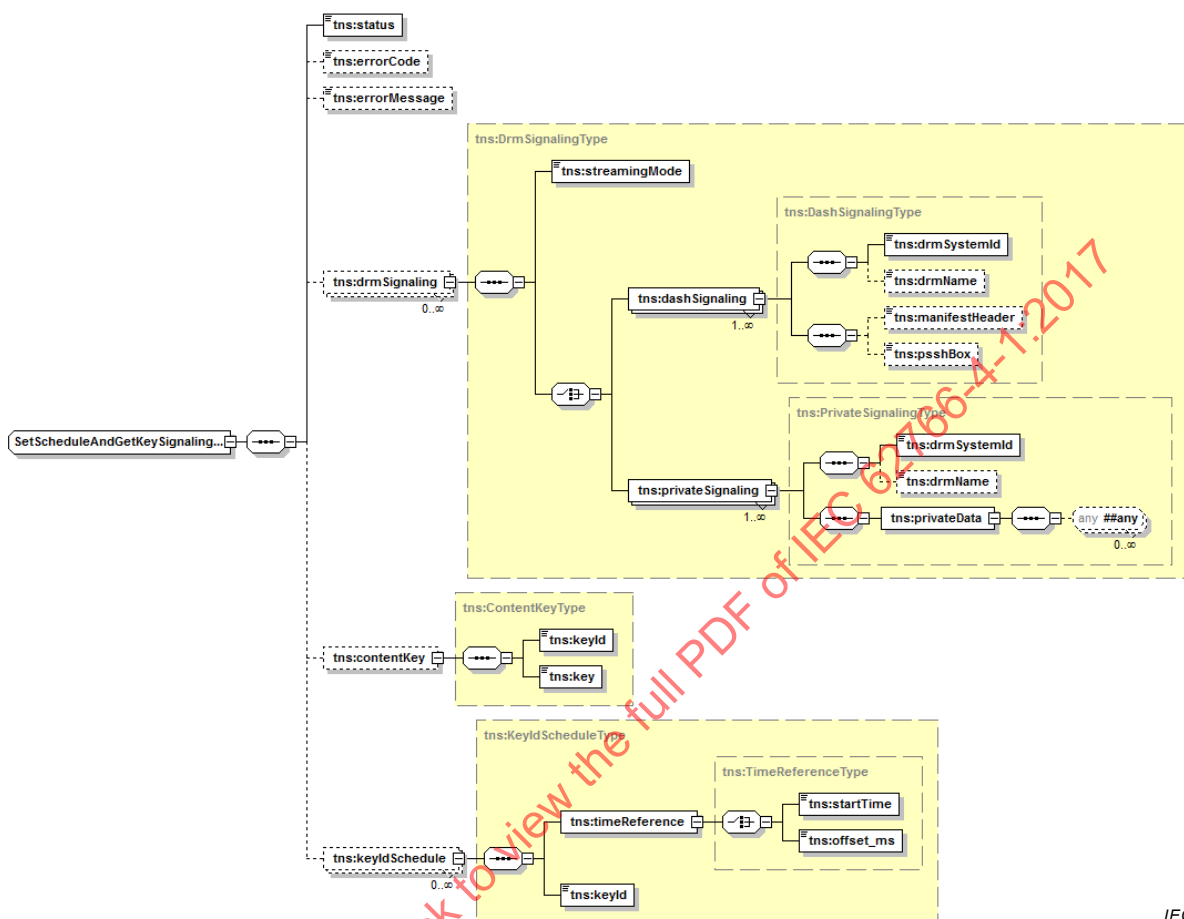
Figure 7 – SetScheduleAndGetKeySignalingrequest schema

Table 113 – SetScheduleAndGetKeySignalingrequestType

Field name	Field type	Multiplicity	Description	Reference
encryptionprofile	EncryptionprofileType	1	Set of properties for the management of the content key and the signaling.	6.7.1.3.6.6
drmInfo	DrmInfoType	0..N	Information for addressed DRM systems. This allows the calling entity to provide a set of DRM systems for which the drmSignaling shall be provided. An empty list means that the called entity manages a DRM list from its own configuration.	6.7.1.3.6.2
schedule	ScheduleType	1..N	The times provided by the Encryption Function for which the Key Management Function has to generate keys.	6.7.1.3.6.11

6.7.1.3.3.3 Output parameter structure definition

Figure 8 shows the XML schema of the SetScheduleAndGetKeySignalingresponse message and Table 114 shows the SetScheduleAndGetKeySignalingresponseType.



IEC

Figure 8 – SetScheduleAndGetKeySignalingresponse schema

Table 114 – SetScheduleAndGetKeySignalingresponseType

Field name	Field type	Multiplicity	Description	Reference
status	StatusType	1	The global status of the request.	6.7.1.3.5.2
errorCode	xs:unsignedInt	0..1	The code of the error in case the status is not OK. Values are specific to the implementation of the called entity. It is omitted if the request succeeds.	
errorMessage	xs:string	0..1	Any additional information related to the status in case of errors. It is omitted if the request succeeds.	
contentKey	ContentKeyType	0..1	The content key to be used to protect the content. It is omitted if the request fails.	6.7.1.3.6.1
drmSignaling	DrmSignalingType	0..N	The signaling information according to every provided streaming mode. This information allows the CSP client to find the key materials required to decrypt a given content, either locally when the key has already been retrieved or remotely when requested for the first time. Each DRM will provide signaling information for a given [content/content key] pair, such as URL of the CSP server or DRM. The DRM information is DRM specific. It is omitted if the request fails.	6.7.1.3.6.4
keyIdSchedule	KeyIdScheduleType	0..N	The dates and times for which keys have been generated, and the keyId of these keys. It is omitted if the request fails.	6.7.1.3.6.7

The called entity generates new keys based on the times provided by the calling entity and provides

- the key (i.e. contentKey element) to use at the requested time (i.e. timeReference element in the schedule element of the request, with isCurrentKey set to true) provided by the calling entity;

- the signaling (i.e. list of drmSignaling elements) to be inserted by the Encryption Function when using the key;
- for the next keys, the key identifiers of the next keys with their times (i.e. keyId and timeReference elements in the schedule element of the request, with isCurrentKey set to false). These times and optionally key identifier should be used to perform the subsequent calls to this method.

One drmSignaling element is provided for every provided streamingMode. For each streamingMode, either a dashSignaling or a privateSignaling element is provided for every addressed DRM. This element holds the drmSystemId and the signaling information (defined by the encryptionprofile). For DASH and CENC, either a manifestheader (XML fragment) or a PSSH box data or both can be provided. The manifestheader is to be inserted in the MPD, the psshBox is to be inserted in the content.

If an error occurs, then no contentKey, no keyIdSchedule, and no drmSignaling elements will be provided.

6.7.1.3.4 setKeyScheduleAndGetSignaling operation

6.7.1.3.4.1 Method and parameters

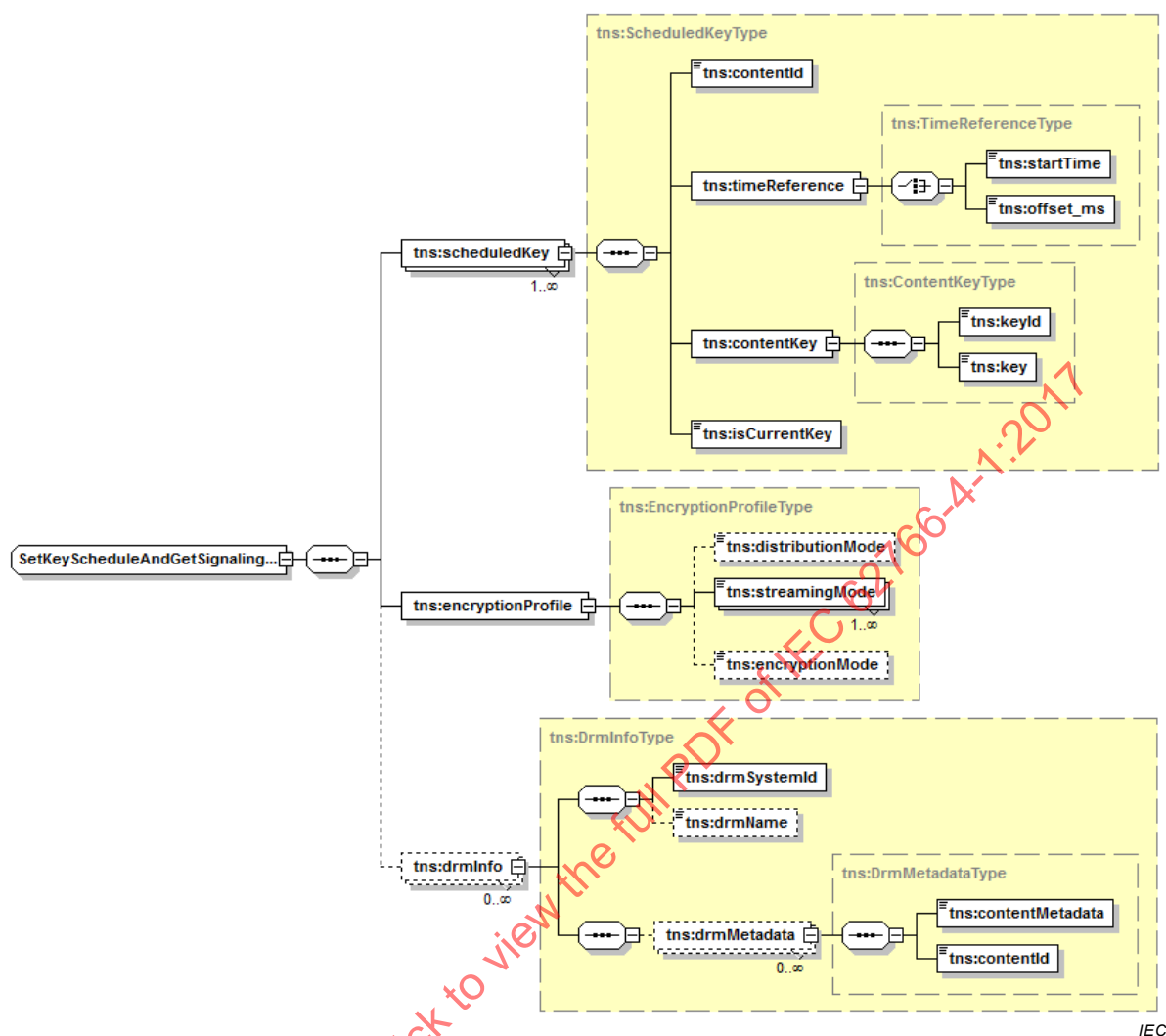
Table 115 shows the setKeyScheduleAndGetSignaling method parameters.

Table 115 – setKeyScheduleAndGetSignaling method parameters

Operation name		Description
setKeyScheduleAndGetSignaling		Operation name
Field name	Field type	Description
SetKeyScheduleAndGetSignalingrequest	SetKeyScheduleAndGetSignalingrequestType	Input parameter
SetKeyScheduleAndGetSignalingresponse	SetKeyScheduleAndGetSignalingresponseType	Output parameter

6.7.1.3.4.2 Input parameter structure definition

Figure 9 shows the XML schema of the SetKeyScheduleAndGetSignalingrequest message and Table 116 shows the SetKeyScheduleAndGetSignalingrequestType.



IEC

Figure 9 – SetKeyScheduleAndGetSignalingrequest schema

Table 116 – SetKeyScheduleAndGetSignalingrequestType

Field name	Field type	Multiplicity	Description	Reference
encryptionprofile	EncryptionprofileType	1	Set of properties for the management of the content key and the signaling.	6.7.1.3.6.6
drmInfo	DrmlInfoType	0..N	Information for addressed DRM systems. This allows the calling entity to provide a set of DRM systems for which the drmSignaling shall be provided. An empty list means that the called entity manages a DRM list from its own configuration.	6.7.1.3.6.2
scheduledKey	ScheduledKeyType	1..N	The keys provided by the Encryption Function.	6.7.1.3.6.10

6.7.1.3.4.3 Output parameter structure definition

Figure 10 shows the XML schema of the SetKeyScheduleAndGetSignalingresponse message and Table 117 shows the SetKeyScheduleAndGetSignalingresponseType.

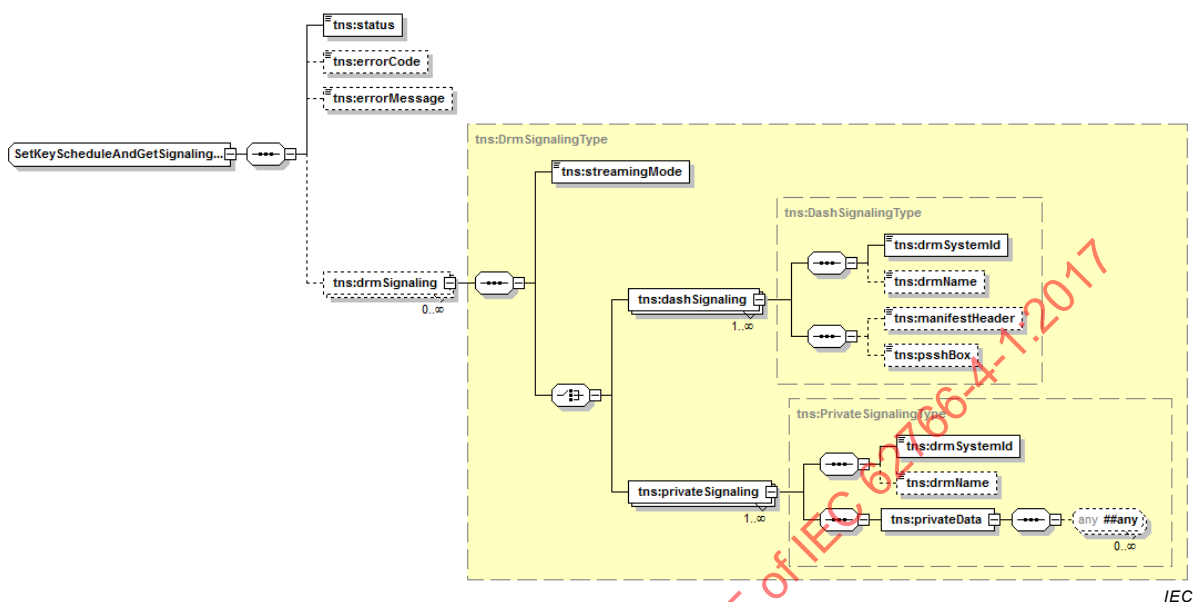


Figure 10 – SetKeyScheduleAndGetSignalingresponse schema

Table 117 – SetKeyScheduleAndGetSignalingresponseType

Field name	Field type	Multiplicity	Description	Reference
status	StatusType	1	The global status of the request.	6.7.1.3.5.2
errorCode	xs:unsignedInt	0..1	The code of the error in case the status is not OK. Values are specific to the implementation of the called entity. It is omitted if the request succeeds.	
errormessage	xs:string	0..1	Any additional information related to the status in case of errors. It is omitted if the request succeeds.	
drmSignaling	DrmSignalingType	0..N	The signaling information according to every provided streaming mode. This information allows the CSP client to find the key materials required to decrypt a given content, either locally when the key has already been retrieved or remotely when requested for the first time. Each DRM will provide signaling information for a given [content/content key] pair, such as URL of the CSP server or DRM. The DRM information is DRM specific. It is omitted if the request failed.	6.7.1.3.6.4

The calling entity provides both the keys and the times to use them. The called entity stores them and provides the signaling (i.e. list of drmSignaling elements) to be inserted by the Encryption Function when using the current key (i.e. scheduleKey element with isCurrentKey set to true).

One drmSignaling element is provided for every provided streamingMode. For each streamingMode, either a dashSignaling or a privateSignaling element is provided for every addressed DRM. This element holds the drmSystemId and the signaling information (defined by encryptionprofile). For DASH and CENC, either a manifestheader (XML fragment) or a PSSH box data or both can be provided. The manifestheader is to be inserted in the MPD, the psshBox is to be inserted in the content.

If an error occurs, then no drmSignaling element will be provided.

6.7.1.3.5 Simple types

6.7.1.3.5.1 UUIDType

This type is used for identifiers that need to be unique in time and space as shown in Table 118, it refers to IETF RFC 4122.

Its main advantage is to not depend on centralized data storage.

Table 118 – UUIDType

Base type	Pattern restriction
xs:token	<code>[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}</code>

6.7.1.3.5.2 StatusType

Table 119 shows the StatusType.

Table 119 – StatusType

Base type	Enumeration	Description
xs:token	OK	The request succeeded.
	ERROR	An occurred at server level. The type of error is provided by the errorCode in the response.

6.7.1.3.6 Data structures

6.7.1.3.6.1 ContentKeyType structure

Table 120 shows the ContentKeyType structure.

Table 120 – ContentKeyType structure

Field name	Field type	Multiplicity	Description
keyId	UUIDType	1	The unique identifier of the content key, provided as a normalized UUID.
key	xs:base64binary	1	The content key used to encrypt content (or part of content in the case of key changes).

6.7.1.3.6.2 DrmInfoType structure

Table 121 shows the DrmInfoType structure.

Table 121 – DrmInfoType structure

Field name	Field type	Multiplicity	Description	Reference
drmSystemId	xs:anyURI	1	The unique identifier of the DRM system, typically a DVB URI or an UUID URI.	
drmName	xs:token	0..1	The informative readable name for the DRM system.	
drmMetadata	DrmMetadataType	0..N	The DRM specific metadata embedding the content related metadata.	6.7.1.3.6.3

6.7.1.3.6.3 DrmMetadataType structure

Table 122 shows the DrmMetadataType structure.

Table 122 – DrmMetadataType structure

Field name	Field type	Multiplicity	Description
contentMetadata	xs:base64binary	1	The content related part of the drmMetadata.
contentId	xs:string	1	Content associated with the metadata.

6.7.1.3.6.4 DrmSignalingType structure

Either one of DASH or HAS field will be provided for a given DRM system, as shown in Table 123.

Table 123 – DrmSignalingType structure

Field name	Field type	Multiplicity	Description	Reference
streamingMode	xs:token	1	Provides the way the content is streamed for the actual drmSignaling element: "DASH" for MPEG Dynamic Http Adaptive Streaming. "HAS" for OIPF Http Adaptive Streaming.	
dashSignaling	DashSignalingType	0..N	DASH signaling as a MPD content protection XML subclause and/or a PSSH box data.	6.7.1.3.6.5
privateSignaling	PrivateSignalingType	0..N	Other signaling, e.g. OIPF HAS.	6.7.1.3.6.9

6.7.1.3.6.5 DashSignalingType structure

Table 124 shows the DashSignalingType structure.

Table 124 – DashSignalingType structure

Field name	Field type	Multiplicity	Description
drmSystemid	xs:anyURI	1	The DRM system id for which the below signaling information applies, typically a DVB URI or an UUID URI.
drmName	xs:token	0..1	Optional readable name of the DRM system.
manifestheader	xs:string	0..1	The content protection subclause to be inserted in the MPD.
psshBox	xs:base64binary	0..1	Pssh signaling information. DRM specific full pssh box binary block.

6.7.1.3.6.6 EncryptionprofileType structure

Table 125 shows the EncryptionprofileType structure.

Table 125 – EncryptionprofileType structure

Field name	Field type	Multiplicity	Description
distributionMode	xs: token	0..1	Either VOD or LIVE, informative: “VOD” for Content On Demand. “LIVE” for Scheduled Content.
streamingMode	xs: token	1..N	Provides the way or ways the content is streamed, helps building the signaling: “DASH” for MPEG Dynamic Http Adaptive Streaming. “HAS” for OIPF Http Adaptive Streaming.
encryptionMode	xs: token	0..1	Encryption method indicator: “DVB-CSA2” for DVB-CSA2. “ATIS-IDSA” for AES 128-bit key using the CBC encryption mode with the IV setting and the residual termination block process as specified in [ATIS-IDSA]. “CENC” for AES 128-bit key in CTR mode (CENC). If not provided then a default value from the system configuration is assumed.

6.7.1.3.6.7 KeyIdScheduleType structure

Table 126 shows the KeyIdScheduleType structure.

Table 126 – KeyIdScheduleType structure

Field name	Field type	Multiplicity	Description	Reference
timeReference	TimeReferenceType	1	The date and time for which a key has been generated.	6.7.1.3.6.12
keyId	UUIDType	1	The unique identifier of the generated key, provided as a normalized UUID.	

6.7.1.3.6.8 PregenerationWindowType structure

Table 127 shows the PregenerationWindowType structure.

Table 127 – PregenerationWindowType structure

Field name	Field type	Multiplicity	Description	Reference
timeReference	TimeReferenceType	1	The date and time at which the requested key and drmSignaling shall apply.	6.7.1.3.6.12
windowDuration_s	xs:unsignedInt	0..1	The width of the time window for which the called entity has to generate keys. If it is not provided then a default value from the system configuration is assumed.	

6.7.1.3.6.9 PrivateSignalingType structure

Table 128 shows the PrivateSignalingType structure.

Table 128 – PrivateSignalingType structure

Field name	Field type	Multiplicity	Description
drmSystemid	xs:anyURI	1	The DRM system id for which the below signaling information applies, typically a DVB URI or an UUID URI.
drmName	xs:token	0..1	Optional readable name of the DRM system.
privateData	xs:any	0..N	Proprietary data block.

6.7.1.3.6.10 ScheduledKeyType structure

Table 129 shows the ScheduledKeyType structure.

Table 129 – ScheduledKeyType structure

Field name	Field type	Multiplicity	Description	Reference
contentId	xs:string	1	Identifier for content, a content being either a scheduled content service, an event on a scheduled content service (i.e. with access criteria specific to the event) or a CoD asset. It is provided by the Scheduling Function (for Scheduled content) or the Content Management Function (for CoD).	
isCurrentKey	xs:boolean	1	Flag to indicate that the requested drmSignaling applies to the period starting according to timeReference. Within a list of elements of type ScheduledKeyType, one and only one occurrence shall have this flag set to true.	
timeReference	TimeReferenceType	1	The time at which the key will start being used.	6.7.1.3.6.12
contentKey	ContentKeyType	1	Content key value and identifier.	6.7.1.3.6.1

6.7.1.3.6.11 ScheduleType structure

Table 130 shows the ScheduleType structure.

Table 130 – ScheduleType structure

Field name	Field type	Multiplicity	Description	Reference
contentId	xs:string	1	Identifier for content, a content being either a scheduled content service, an event on a scheduled content service (i.e. with access criteria specific to the event) or a CoD asset. It is provided by the Scheduling Function (for Scheduled content) or the Content Management Function (for CoD).	
isCurrentKey	xs:boolean	1	Flag to indicate that the requested drmSignaling applies to the period starting according to timeReference. Within a list of elements of type ScheduleType, one and only one occurrence shall have this flag set to true.	
timeReference	TimeReferenceType	1	The date and time at which the requested key and drmSignaling shall apply.	6.7.1.3.6.12
keyId	UUIDType	0..1	Optionally, the unique identifier of an already generated content key to be reused at the time specified, provided as a normalized UUID. If not provided, the called entity decides which key to use based on its schedule and generated keys.	

6.7.1.3.6.12 TimeReferenceType structure

Table 131 shows the TimeReferenceType structure.

Table 131 – TimeReferenceType structure

Field name	Field type	Multiplicity	Description
startTime	xs:dateTime	0..1	For scheduled content, the date and time is an absolute value.
offset_ms	xs:unsignedInt	0..1	For CoD, the time is relative to the beginning of the content. Time is indicated in milliseconds.

6.7.2 XML schema for content preparation

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn:oipf:iptv:KeyAndSignaling:2013"
  targetNamespace="urn:oipf:iptv:KeyAndSignaling:2013"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:simpleType name="StatusType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="OK"/>
      <xs:enumeration value="ERROR"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UUIDType">
    <xs:restriction base="xs:token">
      <xs:pattern value=
        "[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-
F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ContentKeyType">
    <xs:sequence>
      <xs:element name="keyId" type="tns:UUIDType"/>
      <xs:element name="key" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DrmType">
    <xs:sequence>
      <xs:element name="drmSystemId" type="xs:anyURI"/>
      <xs:element name="drmName" type="tns:DrmNameType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DrmSignalingType">
    <xs:sequence>
      <xs:element name="streamingMode" type="xs:token"/>
      <xs:choice>
        <xs:element name="dashSignaling" type="tns:DashSignalingType"
          maxOccurs="unbounded"/>
        <xs:element name="privateSignaling"
          type="tns:PrivateSignalingType"
          maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EncryptionprofileType">
    <xs:sequence>
      <xs:element name="distributionMode" type="xs:token"
        minOccurs="0"/>
      <xs:element name="streamingMode" type="xs:token"
        maxOccurs="unbounded"/>
      <xs:element name="encryptionMode" type="xs:token" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DashSignalingType">
    <xs:complexContent>
      <xs:extension base="tns:DrmType">
        <xs:sequence>
          <xs:element name="manifestheader" type="xs:string"
            minOccurs="0"/>
          <xs:element name="psshBox" type="xs:base64Binary"
            minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="DrmInfoType">
  <xs:complexContent>
    <xs:extension base="tns:DrmType">
      <xs:sequence>
        <xs:element name="drmMetadata" type="tns:DrmMetadataType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="ContentIdType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="PrivateSignalingType">
  <xs:complexContent>
    <xs:extension base="tns:DrmType">
      <xs:sequence>
        <xs:element name="privateData">
          <xs:complexType>
            <xs:sequence>
              <xs:any namespace="##any" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="DrmNameType">
  <xs:restriction base="xs:token"/>
</xs:simpleType>
<xs:complexType name="DrmMetadataType">
  <xs:sequence>
    <xs:element name="contentMetadata" type="xs:base64Binary"/>
    <xs:element name="contentId" type="tns:ContentIdType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ScheduledKeyType">
  <xs:sequence>
    <xs:element name="contentId" type="tns:ContentIdType"/>
    <xs:element name="timeReference" type="tns:TimeReferenceType"/>
    <xs:element name="contentKey" type="tns:ContentKeyType"/>
    <xs:element name="isCurrentKey" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="GetKeyScheduleSignalingrequestType">
  <xs:sequence>
    <xs:element name="contentId" type="tns:ContentIdType"/>
    <xs:element
      name="encryptionprofile"
type="tns:EncryptionprofileType"/>
    <xs:element name="drmInfo" type="tns:DrmInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="pregenerationWindow"
      type="tns:PregenerationWindowType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="SetKeyScheduleAndGetSignalingrequestType">
  <xs:sequence>
    <xs:element name="scheduledKey" type="tns:ScheduledKeyType"

```

```

        maxOccurs="unbounded"/>
        <xs:element name="encryptionprofile"
type="tns:EncryptionprofileType"/>
        <xs:element name="drmInfo" type="tns:DrmInfoType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="SetScheduleAndGetKeySignalingrequestType">
    <xs:sequence>
        <xs:element name="schedule" type="tns:ScheduleType"
maxOccurs="unbounded"/>
        <xs:element name="encryptionprofile"
type="tns:EncryptionprofileType"/>
        <xs:element name="drmInfo" type="tns:DrmInfoType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="TimeReferenceType">
    <xs:choice>
        <xs:element name="startTime" type="xs:dateTime"/>
        <xs:element name="offset_ms" type="xs:unsignedInt"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="ScheduleType">
    <xs:sequence>
        <xs:element name="contentId" type="tns:ContentIdType"/>
        <xs:element name="isCurrentKey" type="xs:boolean"/>
        <xs:element name="timeReference" type="tns:TimeReferenceType"/>
        <xs:element name="keyId" type="tns:UIDType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="SetKeyScheduleAndGetSignalingresponseType">
    <xs:sequence>
        <xs:element name="status" type="tns:StatusType"/>
        <xs:element name="errorCode" type="xs:unsignedInt" minOccurs="0"/>
        <xs:element name="errorMessage" type="xs:string" minOccurs="0"/>
        <xs:element name="drmSignaling" type="tns:DrmSignalingType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="SetScheduleAndGetKeySignalingresponseType">
    <xs:sequence>
        <xs:element name="status" type="tns:StatusType"/>
        <xs:element name="errorCode" type="xs:unsignedInt" minOccurs="0"/>
        <xs:element name="errorMessage" type="xs:string" minOccurs="0"/>
        <xs:element name="drmSignaling" type="tns:DrmSignalingType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="contentKey" type="tns:ContentKeyType"
minOccurs="0"/>
        <xs:element name="keyIdSchedule" type="tns:KeyIdScheduleType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="GetKeyScheduleSignalingresponseType">
    <xs:sequence>
        <xs:element name="status" type="tns:StatusType"/>
        <xs:element name="errorCode" type="xs:unsignedInt" minOccurs="0"/>
        <xs:element name="errorMessage" type="xs:string" minOccurs="0"/>
        <xs:element name="drmSignaling" type="tns:DrmSignalingType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="contentKey" type="tns:ContentKeyType"
minOccurs="0"/>

```

```

    <xs:element name="timeReference" type="tns:TimeReferenceType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="KeyIdScheduleType">
  <xs:sequence>
    <xs:element name="timeReference" type="tns:TimeReferenceType"/>
    <xs:element name="keyId" type="tns:UUIDType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PregenerationWindowType">
  <xs:sequence>
    <xs:element name="timeReference" type="tns:TimeReferenceType"/>
    <xs:element name="windowDuration_s" type="xs:unsignedInt"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="GetKeyScheduleSignalingrequest"
  type="tns:GetKeyScheduleSignalingrequestType"/>
<xs:element name="SetKeyScheduleAndGetSignalingrequest"
  type="tns:SetKeyScheduleAndGetSignalingrequestType"/>
<xs:element name="SetScheduleAndGetKeySignalingrequest"
  type="tns:SetScheduleAndGetKeySignalingrequestType"/>
<xs:element name="SetKeyScheduleAndGetSignalingresponse"
  type="tns:SetKeyScheduleAndGetSignalingresponseType"/>
<xs:element name="SetScheduleAndGetKeySignalingresponse"
  type="tns:SetScheduleAndGetKeySignalingresponseType"/>
<xs:element name="GetKeyScheduleSignalingresponse"
  type="tns:GetKeyScheduleSignalingresponseType"/>
</xs:schema>

```

The following WEB service is introduced according to the protocols defined in 6.7.1.3.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wsdl:definitions
  name="IKeyAndSignaling"
  targetNamespace="urn:oipf:iptv:KeyAndSignaling:2013"
  xmlns:tns="urn:oipf:iptv:KeyAndSignaling:2013"
  xmlns:defs="urn:oipf:iptv:KeyAndSignaling:2013"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

  <wsdl:import namespace="urn:oipf:iptv:KeyAndSignaling:2013"
    location="KeyAndSignaling.wsdl" />

  <wsdl:binding name="KeyAndSignalingSoapBinding"
    type="defs:KeyAndSignalingPortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>

    <wsdl:operation name="getKeyScheduleSignaling">
      <soap:operation soapAction="ws-keyAndSignaling"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="setKeyScheduleAndGetSignaling">

```

```

    <soap:operation soapAction="ws-keyAndSignaling"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="setScheduleAndGetKeySignaling">
    <soap:operation soapAction="ws-keyAndSignaling"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

</wsdl:binding>

<wsdl:service name="KeyAndSignalingService">
  <wsdl:port name="KeyAndSignalingSoapPort"
    binding="tns:KeyAndSignalingSoapBinding">
    <soap:address location="ws-keyAndSignaling" />
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>

```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wsdl:definitions
  name="KeyAndSignaling"
  targetNamespace="urn:oipf:iptv:KeyAndSignaling:2013"
  xmlns:tns="urn:oipf:iptv:KeyAndSignaling:2013"
  xmlns:kas="urn:oipf:iptv:KeyAndSignaling:2013"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <wsdl:types>
    <xs:schema>
      <xs:import namespace="urn:oipf:iptv:KeyAndSignaling:2013"
        schemaLocation="iptv-KeyAndSignaling.xsd"/>
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="GetKeyScheduleSignalingrequest">
    <wsdl:part
      element="kas:GetKeyScheduleSignalingrequest"/>
    </wsdl:part>
  </wsdl:message>

  <wsdl:message name="GetKeyScheduleSignalingresponse">
    <wsdl:part
      element="kas:GetKeyScheduleSignalingresponse"/>
    </wsdl:part>
  </wsdl:message>

  <wsdl:message name="SetKeyScheduleAndGetSignalingrequest">
    <wsdl:part
      element="kas:SetKeyScheduleAndGetSignalingrequest"/>
    </wsdl:part>
  </wsdl:message>

  <wsdl:message name="SetKeyScheduleAndGetSignalingresponse">

```

```

    <wsdl:part
element="kas:SetKeyScheduleAndGetSignalingresponse"/>
    </wsdl:message>

    <wsdl:message name="SetScheduleAndGetKeySignalingrequest">
        <wsdl:part
element="kas:SetScheduleAndGetKeySignalingrequest"/>
        </wsdl:message>
    <wsdl:message name="SetScheduleAndGetKeySignalingresponse">
        <wsdl:part
element="kas:SetScheduleAndGetKeySignalingresponse"/>
        </wsdl:message>

<wsdl:portType name="KeyAndSignalingPortType">
    <wsdl:operation name="getKeyScheduleSignaling">
        <wsdl:input message="tns:GetKeyScheduleSignalingrequest"
            name="GetKeyScheduleSignalingInput"/>
        <wsdl:output message="tns:GetKeyScheduleSignalingresponse"
            name="GetKeyScheduleSignalingOutput"/>
    </wsdl:operation>
    <wsdl:operation name="setKeyScheduleAndGetSignaling">
        <wsdl:input message="tns:SetKeyScheduleAndGetSignalingrequest"
            name="SetKeyScheduleAndGetSignalingInput"/>
        <wsdl:output message="tns:SetKeyScheduleAndGetSignalingresponse"
            name="SetKeyScheduleAndGetSignalingOutput"/>
    </wsdl:operation>
    <wsdl:operation name="setScheduleAndGetKeySignaling">
        <wsdl:input message="tns:SetScheduleAndGetKeySignalingrequest"
            name="SetScheduleAndGetKeySignalingInput"/>
        <wsdl:output message="tns:SetScheduleAndGetKeySignalingresponse"
            name="SetScheduleAndGetKeySignalingOutput"/>
    </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```

7 SIP and SIP/SDP

7.1 General

Within the architecture, there are several interfaces that support the SIP/SDP protocol. They can be functionally grouped as follows:

- between the OITF and the IG over the HNI-IGI interface;
- between the IG and the IMS core network covering the UNIS-8 reference point (residential LAN to core network interface);
- internally within the IMS network in support of various services covering reference points NPI-3, NPI-4 and NPI-30;
- between the IMS and the content delivery network covering reference points NPI-19, and NPI-25.

Subclauses 7.2 and 7.3 discuss these interfaces in more detail. Subclause 7.2 covers the SIP/SDP interface in the residential LAN (bullet 1 above), and subclause 7.3 covers the SIP/SDP interfaces within the managed network relying on SIP (last 3 bullets above). Interface details are specified for each service within the sub-subclauses.

7.2 SIP/SDP reference points within the provider network

7.2.1 General

Subclause 7.2 defines the protocol for the use of SIP and SIP/SDP within managed networks relying on IMS over the following reference points:

- NPI-19;
- NPI-26;
- NPI-30;
- NPI-25;
- NPI-3;
- NPI-4;
- UNIS-8.

The usage of SIP/SDP for all reference points apart from UNIS-8 is applicable to both the SIP option and the HTTP option for the HNI-IGI interface.

The usage of SIP/SDP for the UNIS-8 interface is described in 7.3.

7.2.2 Generic handling for SIP requests/responses

7.2.2.1 Locating the target OITF device for incoming requests

For all incoming SIP requests to the IG that are standalone or dialog initiating, the IG shall compare the information in the Accept-Contact header in the incoming SIP request, if available, against stored feature tags for the user to select the appropriate OITF device.

7.2.2.2 Handling of incoming and outgoing SIP requests

The IG shall support the procedures specified in ETSI TS 124 503 for originating and terminating sessions.

7.2.3 Protocols for IPTV service functions

7.2.3.1 Multicast content streaming with SIP session management

7.2.3.1.1 Protocol over UNIS-8

7.2.3.1.1.1 Retrieval of bandwidth parameter for FCC/RET enabled multicast content service

When a request to send SIP OPTIONS is received from the OITF, the IG shall use the mapping specified in 6.3.1.3.

When the final response to the SIP OPTIONS message is received from the network as a SIP 200 OK including the SDP, the IG shall forward this information to the OITF.

The information required in the returned SDP to complete the missing parameters in the SDP offer is

- the total bandwidth for the multicast content service.

7.2.3.1.1.2 Session initiation and modification

Upon receiving a request from the OITF for the initiation of a multicast content service session (see 6.3.1.2), the IG shall generate an initial INVITE request as specified in ETSI TS 124 503 for originating sessions.

The IG shall forward any received SIP response to the OITF including the information in the SDP.

If the IG receives a 488 error code with warning 370 Insufficient Bandwidth, the IG shall send an error message to the OITF.

The session modification procedure is handled by the IG in the same way as a session initiation. See 6.3.1.4.

7.2.3.1.1.3 Session termination

On receiving a request from the OITF for the termination of a multicast content service session (see 6.3.1.5), the IG shall generate a BYE request as specified in ETSI TS 124 503 for originating sessions.

Alternatively, on receipt of a BYE request from the IPTV control FE, the IG shall forward the request to the OITF as a response to a PENDING_IG request (see 6.6). The behaviour of the UNIS-8 part of the IG shall comply with the procedure specified in ETSI TS 124 503 for terminating UA.

7.2.3.1.1.4 Content reporting and management of content reporting

Upon receiving a request from the OITF for reporting the watched content for a multicast content service session (see 6.3.1.7.1), the IG shall generate a SIP INFO including the Content-Reporting Info Package.

The IG shall forward any received SIP response to the OITF.

Upon receipt of a SIP UPDATE by the IG with an empty Recv-Info header or the unwillingness to receive the Content-Reporting Info Package from the IPTV control FE, to stop reporting watched content, the IG shall forward the SIP UPDATE to the OITF in an HTTP 200 OK response. The IG shall wait for the response from the OITF to forward it to the IPTV control FE.

Upon receipt of an SIP UPDATE, including the Recv-Info header set to Content-Reporting Info Package by the IG from the IPTV control FE, to start reporting watched content, the IG shall forward the SIP UPDATE to the OITF in an HTTP 200 OK response. The IG shall wait for the response from the OITF to forward it to the IPTV control FE.

7.2.3.1.1.5 User-initiated activation/deactivation of network-based multicast content streaming time shift

The following procedure is supported in the IG on UNIS-8 in support of user-initiated activation/de-activation of multicast content service time shift:

- upon receiving a request from the OITF for session modification for an activation/de-activation of scheduled content time shift, the IG shall generate a SIP re-INVITE as per ETSI TS 124 503;
- the IG shall forward the received response to the OITF including any SDP information.

The IG shall generate SIP ACK when requested by the OITF.

7.2.3.1.2 Protocol over NPI-4

7.2.3.1.2.1 Retrieval of bandwidth parameter for FCC/RET enabled multicast content service

The SIP OPTIONS message shall conform to ETSI TS 124 503 and shall be forwarded through the ASM, to the IPTV control FE. If the request URI in the incoming SIP OPTIONS

request is set to the PSI for the multicast content service, the IPTV control FE shall, after validating the SIP OPTIONS, perform the following:

- If the incoming SIP OPTIONS includes the a attribute a=rtcp-fb:<fmt> nack under the m-line, the IPTV control FE shall calculate the total bandwidth for the highest multicast content and the bandwidth for RET and shall return the information included in the body of an SIP 200 OK, and which includes the same m-line copied from the incoming request with an additional b-line that includes the total bandwidth.
- If the incoming SIP OPTIONS includes the a attribute a=rtcp-fb:<fmt> nack rai under the m-line, the IPTV control FE shall calculate the total bandwidth for the highest multicast content and the bandwidth for FCC and shall return the information included in the body of a SIP 200 OK, and which includes the same m-line copied from the incoming request with an additional b-line that includes the total calculated bandwidth.
- If the incoming SIP OPTIONS includes the a attribute a=rtcp-fb:<fmt> nack and the a attribute a=rtcp-fb:<fmt> nack rai under the m-line, the IPTV control FE shall calculate the total bandwidth for the highest multicast content and the bandwidth for FCC and RET service combined, and shall return the information included in the body of a SIP 200 OK, and which includes the same m-line copied from the incoming request with an additional b-line that includes the total calculated bandwidth.

7.2.3.1.2.2 Session initiation

The IPTV control FE shall support the procedures specified in ETSI TS 183 063, 5.3.1.1.

The IPTV control FE shall support the procedures specified in ETSI TS 124 503 that are applicable to an AS acting as a terminating SIP UA.

Upon receipt of a SIP INVITE request, the IPTV control FE shall examine the request-URI to determine that it is a multicast content service session initiation request. The IPTV control FE shall use the IPTV Subscription profile to check the service rights for the requested broadcast service packages and multicast addresses. The IPTV control FE shall examine the SDP offer parameters, as defined in ETSI TS 183 063, 5.3.1.1.

If the SDP parameters are validated successfully, the IPTV control FE shall respond as defined in ETSI TS 183 063, 5.3.1.1.

If the SDP also contains m-lines for Network Generated Notification stream, the IPTV control FE shall further check the service rights for the requested notification service, using the IPTV Subscription.

If no bc_service_package attributes are included in the SDP offer, the IPTV control FE shall include in the SDP answer one or more a=bc_service_package attributes, except if it knows that the RACS is or shall be pre-provisioned with the list of subscribed channels and if all the subscribed channels are allowed for the session. In this case, the inclusion of a=bc_service_package is OPTIONAL.

The service packages shall be populated according to the IPTV Subscription profile to indicate the service packages and scheduled content services.

7.2.3.1.2.3 Session modification

The IPTV control FE shall support the procedures specified in ETSI TS 183 063, 5.3.1.2. Network initiated session modification does not apply.

7.2.3.1.2.4 Session termination

The IPTV control FE shall support the procedures specified in ETSI TS 183 063, 5.3.1.4.

7.2.3.1.3 Content reporting and management of content reporting

Upon receipt of SIP INFO including the Info Package Content Reporting for reporting the watched content for a multicast content service session, the IPTV control FE shall examine the body.

If the body is successfully validated as compliant to Annex D of 3GPP TS 26.237, then the IPTV control FE shall respond to the request with a SIP 200 OK, otherwise an appropriate error message shall be returned.

If the IPTV control FE desires to instruct the OITF to stop reporting watched content, it shall send an SIP UPDATE to the IG and where the Recv-Info header is set to remove support for the reception of the Content-Reporting Info Package and shall wait for the response.

If the IPTV control FE desires to instruct the OITF to start reporting watched content, it shall send a SIP UPDATE to the IG and where the Recv-Info header is set to indicate support for the reception of Content-Reporting Info Package and shall wait for the response.

7.2.3.1.4 User-initiated activation/deactivation of network-based multicast content streaming time shift

Upon receipt of a SIP re-INVITE related to an activation/de-activation of a time shift session, the IPTV control FE shall verify that it does hold a state for the session to be modified.

If the IPTV control FE does not hold a state for the session, or if the multipart of the message body including, both, the SDP parameters, and the XML schema for the OITF-IPTV Commands as per 6.3.1.8.3 are not successfully validated, the IPTV control FE shall reject the request with a 403 error code.

If the request is for the activation of time shift, the IPTV control FE shall perform the procedure defined in 7.2.3.2.2, and shall return to the IG the appropriate response including all mandatory parameters as per the said procedure (note that if the multicast content stream is not stored in the network, the procedure in 7.2.3.2.2 shall fail and an appropriate error message shall be returned to the user).

If the request is for de-activation of time shift, the IPTV control FE shall perform the procedure defined in 7.2.3.2.3. Following the successful completion of the said procedure, the IPTV control FE shall return a SIP 200 OK to the IG.

7.2.3.2 Unicast content streaming with SIP session management

7.2.3.2.1 Retrieving missing parameters in the SDP prior to session setup using SIP OPTIONS

7.2.3.2.1.1 Protocol over UNIS-8

When a request to send a SIP OPTIONS is received from the OITF, the IG shall use the mapping specified in 6.3.2.1.1.

When the final response to the SIP OPTIONS message is received from the network as a SIP 200 OK including the RTSP SDP, the IG shall forward this information to the OITF.

The information required in the returned SDP to complete the missing parameters in the SDP offer is

- FEC information including bandwidth for FEC streams,
- Transport protocol.

7.2.3.2.1.2 Protocol over NPI-4, NPI-19, NPI-26

The SIP OPTIONS message shall conform to ETSI TS 124 503 and shall be forwarded through the ASM to the IPTV Control. If, the request URI in the incoming SIP OPTIONS request is set to the PSI for the unicast content streaming service, the IPTV Control shall, after validating the request, forward the SIP OPTIONS to the CDN controller FE and to the appropriate cluster controller, in the same way as for the INVITE message.

In certain cases, the CDN controller may forward the SIP OPTIONS message to a default cluster controller.

On receiving the SIP OPTIONS message, the cluster controller shall issue an RTSP DESCRIBE to the CDF. In certain cases, the cluster controller may issue an RTSP DESCRIBE to a default CDF.

The DESCRIBE message shall conform to the format defined by ETSI TS 102 034.

The XML description that complies with ETSI TS 102 034 included in the RTSP 200 OK response received from the CDF shall be converted by the cluster controller to the SDP body in a SIP 200 OK response to the OPTIONS message. The SIP 200 OK message shall be forwarded all the way back to the IG.

The explicit mapping between the XML description and the SDP is not subject to formal specification.

This is the only case for which an OPTIONS message will be sent to a cluster controller.

NOTE If, in a future release, other reasons warrant that the cluster controller receive the OPTIONS message, then support for discrimination between the various reasons for sending the OPTION will be required.

7.2.3.2.2 Procedure for unicast service session initiation

7.2.3.2.2.1 Session initiation

The IG shall support the procedures specified in ETSI TS 124 503 for initiating unicast sessions.

On receiving a request for a unicast session initiation from the OITF, the IG shall generate an initial INVITE request as specified in ETSI TS 124 503 (for an originating UA). See 6.3.2.1.2.

See example messages in A.1.

7.2.3.2.2.2 Protocol over NPI-4

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

When receiving any SIP request, the IPTV control FE shall examine the request to see if it is compatible with the user's subscription profile (e.g. parental control level). If the user is not allowed to initiate a session for the requested content, the IPTV control FE shall reply with an appropriate SIP error response. If the user is allowed to initiate the session, the IPTV control FE shall forward the SIP INVITE to a default CDN controller.

The IPTV control function shall not change the user-part of the To header in order to retain the content-id in the INVITE request.

Note that this does not apply for the case when the session is related to a user-activated time shift. In case of user-activated time shift, the IPTV control FE shall change the To field to include the requested content ID that corresponds to the shifted multicast content. When the

final response is received, the IPTV control FE shall restore the To field to its original value. In that respect, the IPTV control FE maintains this information in the session state as long as the session is alive.

7.2.3.2.2.3 Protocol over NPI-19

The CDN controller FE shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

When receiving the SIP INVITE from the IPTV control FE via the authentication and session Management FE through the NPI-19 reference point, the CDN controller shall check the content id in the user part of the “To:” header as well as the “From:” and “Via:” fields to determine the most appropriate cluster controller FE to serve the User's request.

Once the appropriate cluster controller FE is selected, the Content Delivery Network Controller FE shall forward the SIP INVITE to it by changing the “request-URI” accordingly.

The CDN controller shall not forward 301 or 302 responses from the cluster controller to the IPTV control function. The CDN controller shall take one of the following actions on receiving a 301 or 302 response from the cluster controller:

- cancel the transaction;
- forward to another cluster controller;
- forward to the suggested CC as indicated in the 301/302 response;
- forward to another CDN controller.

7.2.3.2.2.4 Protocol over NPI-25

On receiving the request from the IPTV control function, the CDN controller may decide to forward the request to another CDN controller. In this case, it changes the “request-URI” accordingly.

7.2.3.2.2.5 Protocol over NPI-26

The cluster controller FE shall support the procedures specified in ETSI TS 124 503 as applicable to a terminating UA.

When receiving a unicast content streaming session initiation SIP request from the CDN controller, the cluster controller shall examine the content identifier present in the user-part of the “To:” header and the media parameters in the received SDP offer and then choose the CDF.

If the requested content is not managed by this cluster controller, the cluster controller shall return a 301 response, or a 302 response for any other reasons (e.g. load-balancing). The cluster controller may indicate one or more cluster controller addresses in the contact header as indicated in IETF RFC 3261.

If the request is not acceptable to the cluster controller, it shall reply with an appropriate SIP error response.

The cluster controller shall reply with an appropriate SIP error response if the request is acceptable to the cluster controller but none of the content delivery functions can handle the offer.

If the request is acceptable to the cluster controller and a CDF can handle the request, the cluster controller shall initiate an RTSP session using the RTSP SETUP message to the chosen CDF to determine its server ports and the RTSP session ID.

Following the successful conclusion of the RTSP session setup, the cluster controller allocates an RTSP server port, binds it to the CDF RTSP server port and answers with a SIP 200 OK, including the SDP answer.

The SDP parameters for the RTSP channel shall be set as follows:

- An m-line for an RTSP stream with the format: m=<media> <port> <transport> <fmt> (ex. m=application 554 tcp iptv_rtsp)
- The <media> field shall have a value of “application”.
- The <port> field shall be setup according to IETF RFC 4145. The port number shall be set to the port allocated by the cluster controller.
- The <transport> field shall be identical to the one received in the SDP offer in the initial INVITE.
- The <fmt> field shall be identical to the one received in the SDP offer in the initial INVITE.
- A c-line shall include the network type with the value set to “IN”, the address type set to “IP4” and the IP address for the RTSP commands. (example: c=IN IP4 <RTSP IP address>)
- An “a=setup” attribute shall be present and set to “passive”, indicating that the connection is initiated by the other endpoint (OITF), as defined in IETF RFC 4145. (example: a=setup:passive)
- An “a= connection” attribute shall be present and set to “new” as defined in IETF RFC 4145. (ex: a=connection:new)
- One or more a=fmtp lines representing RTSP specific attributes set as follows:
 - A “fmtp:iptv_rtsp h-uri” attribute shall be set to the RTSP URI of the cluster controller to be used in the RTSP requests. The h-uri can be in the form of an absolute or relative URI. If an absolute URI is specified, then it shall be used in subsequent RTSP requests. If a relative URI is specified in the form of a media path, then the RTSP absolute URI could be constructed by the OITF using the IP Address (from c-line) and port (from m-line) as the base followed by h-uri value for the media path. (i.e. fmtp:iptv_rtsp h-uri=<request-uri>)
 - An absolute URI shall have precedence over a c-line if the latter is provided.
 - The cluster controller shall include a “fmtp:iptv_rtsp h-session” attribute representing the session-id of the RTSP session to be used by the OITF during media control. Optionally, a timeout parameter may be specified with a numeric timeout interval in seconds for keep-alive (refer to 8.2.1.2.4). If the timeout parameter is not specified, then a default value of 60 s shall be used (refer to 12.37 of IETF RFC 2326) (i.e. a=fmtp:iptv_rtsp h-session=<rtsp-session>[; timeout=<timeout>])
 - Note that if both RTP and RTCP are used in the session, the RTSP server (CDF) can use the received RTCP messages as an indication that the OITF is still connected to the session. This avoids requiring explicit RTSP keep-alive signalling. The RTSP server can easily associate the RTCP messages to the RTSP session-ID using the RTCP message transport address and the SSRC of the media source. If this method is used and no RTCP messages are received after the default timeout period, the RTSP server may tear down the session. Details of this methodology are explained below in the b=RR:<bandwidth-value> line.
- An m-line for the actual content which indicates the type of the media, the transport protocol and the port of the related content delivery channel from the response message for the RTSP DESCRIBE. If a fmt parameter is in the SDP offer, it shall be completed with the supported format by the CDF.
- A c-line shall include the network type with the value set to IN, the address type set to IP4 and the unicast address of the stream related to the content delivery channel. (i.e. c=IN IP4 <IP_ADDRESS>)

- A b-line shall contain the proposed “session bandwidth” for the COD media stream. Note that this bandwidth value includes the IP and UDP headers (see 6.2 of IETF RFC 3550) (e.g. b=AS:64, indicating 64 kbit/s)
- An a-line with a “sendonly” (e.g. a=sendonly)
- A b-line, b=RR:<bandwidth-value>, specifying the agreed bandwidth value (in kbit/s) the OITF shall allocate for sending Receiver Reports (RR) in the COD session.
- The cluster controller may set the bandwidth value in the answer to zero (0) even if a non-zero value is requested by the OITF. In the event a non-zero value is requested by the OITF, the cluster controller shall not change the proposed bandwidth value to a different non-zero value as this could force the OITF to use a bandwidth value it may not be able to allocate causing COD session failure.
- Optionally, a b-line b=RS:<bandwidth-value> specifying the amount of bandwidth (in kbit/s) that the COD session senders (in this case only one COD server) shall allocate for RTCP Sender Reports (SR). This value may be set to zero (0), b=RS:0. Setting this value to zero (0) is not recommended if several streams will be synchronised.

7.2.3.2.3 Session termination

Session termination for unicast content streaming shall follow the same SIP procedures as session termination for the multicast content streaming service. See 7.2.3.1.1.3.

7.2.3.3 Forced play-out control with SIP session management Forced play-out controlled by the cluster controller

7.2.3.3.1 General

Client-based play out control is described in IEC 62766-2-1:2017, 4.1 and 4.2, and in IEC 62766-7:2017, Clause 6. The concept there is applicable to downloaded, streamed or stored content where navigation constraints are embedded into the content, i.e. into MP4 files or MPEG-2 transport streams. IEC 62766-5-1 contains language that allows a DAE application to capture navigation input and programmatically decide to honour or ignore the navigation commands, thus providing another way to restrict navigation when content is consumed in a DAE application.

7.2.3.3.2 Session initiation

Forced Play Out Control session initiation over UNIS-8 is the same procedure as described in 7.2.3.2.2.1.

7.2.3.3.3 Protocol over NPI-4

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

When receiving any SIP request, the IPTV control FE shall examine the request to see if it is compatible with the user's subscription profile (e.g. parental control level). If the user is not allowed to initiate a session for the requested content, the IPTV control FE shall reply with an appropriate SIP error response. If the user is allowed to initiate the session, the IPTV control FE shall interact with the IPTV application to acquire the Forced Play Out control policy per the content identity and/or user's subscription information.

The IPTV control FE shall add the policy to the SIP INVITE received, and forward the updated SIP INVITE to a default CDN controller. The IPTV control function shall not change the user-part of the To header in order to retain the content-id in the INVITE request.

The IPTV control function shall not change the existing SDP parameters for the RTSP content control channel and content delivery channel which specified in 6.3.2.1.2. Moreover, an

additional SDP parameter for the RTSP content control channel to describe Forced Play Out control policy shall be included as follows:

- one or more “a=unsupport” attributes (see below) as defined in 7.2.3.3.7.

Note that this does not apply for the case when the session is related to a user-activated time shift. In case of user-activated time shift, the IPTV control FE shall change the To field to include the requested content ID that corresponds to the shifted multicast content. When the final response is received, the IPTV control FE shall restore the TO field to its original value. In that respect, the IPTV control FE maintains this information in the session state as long as the session is alive.

7.2.3.3.4 Protocol over NPI-19

Protocol for Forced Play Out Control over NPI-19 is the same as described in 7.2.3.2.2.3.

7.2.3.3.5 Protocol over NPI-25

On receiving the request from the IPTV control function, the CDN controller may decide to forward the request to another CDN controller. In this case, the IPTV control function changes the “request-URI” accordingly.

7.2.3.3.6 Protocol over NPI-26

The protocol for Forced Play Out Control over NPI-26 is similar to that described in 7.2.3.2.2.5.

After receiving the session initiation request, if the policy is presented in the request, the cluster controller shall also store the Forced Play Out Control policy and then choose the CDF.

7.2.3.3.7 Forced play-out control policy attribute

The format of the “a=unsupport” attribute is as follows:

```
a=unsupport: < RTSP_Operation > < Restriction_Rule >
where
<RTSP_Operation>::= is the RTSP operation that is not permitted by the
policy, e.g. fast forward.
<Restriction_Rule>::= <time-range-list> | <ContentID-list>
<ContentID-list>::= ContentID_list: <contentID> {"|" <contentID>}
<ContentID >::= is the identifier of the media content
<time-range-list>::= <npt-range-list> | < utc-range-list> | <smpte-
range-list>
<npt-range-list>::= npt_range_list: <npt-range> {"|" <npt-range>}
<utc-range-list>::= utc_range_list: <utc-range> {"|" <utc-range>}
<smpte-range-list>::= smpte_range_list: <smpte-range> {"|" <smpte-
range>}
```

7.2.3.4 Purchase of digital media service using SIP purchase request for digital media

7.2.3.4.1 Protocol over UNIS-8

Upon receiving a purchase request from the OITF, the IG shall send a SIP INFO request, including the Digital-purchase info package, IPTV control FE (via the ASM FE).

The IG shall forwards any received response to the OITF.

Upon receipt of a SIP UPDATE by the IG with an empty Recv-Info header or unwillingness to receive the Digital-purchase info package from the IPTV control FE, to stop sending Digital-

Media-Purchase request, the IG shall forward the SIP UPDATE to the OITF in an HTTP 200 OK response. The IG shall wait for the response from the OITF to forward it to the IPTV control FE.

Upon receipt of a SIP UPDATE, including the Recv-Info header set to support the Digital-purchase Info Package, by the IG from the IPTV control FE, to start sending Digital-Media-Purchase request, the IG shall forward the SIP UPDATE to the OITF in an HTTP 200 OK response. The IG shall wait for the response from the OITF to forward it to the IPTV control FE.

Note that the IG shall be stateful to the SIP UPDATE messages so that it does comply to IPTV control FE in case an OITF makes an illegal request.

7.2.3.4.2 Protocol over NPI-4

Upon receipt of SIP INFO request from the IG including the Digital-purchase info package, the IPTV control FE shall examine the body. If the body is not successfully validated as compliant with 6.3.5.9, the IPTV control FE shall issue an appropriate error message. Otherwise, the IPTV control FE shall initiate a purchase request with the Charging FE.

Following the successful completion of the purchase transaction, the IPTV control FE shall inform the IPTV applications FE so it can update the user profile, and shall wait for a response from the IPTV applications FE before returning a response to the IG.

If the IPTV control FE desires to instruct the OITF to start sending Digital-Media-Purchase request, it shall send a SIP UPDATE to the IG and where the Recv-Info is set to indicate support for the reception of Digital-purchase Info Package and shall wait for the response.

If the IPTV control FE desires to instruct the OITF to stop sending Digital-Media-Purchase request, it shall send a SIP UPDATE to the IG and where the Recv-Info is empty or removes support for the reception of the Digital-purchase Info Package and shall wait for the response.

7.2.3.5 Pay-per-view multicast content service with SIP session management

7.2.3.5.1 Protocol over UNIS-8

7.2.3.5.1.1 PPV service initiation without existing multicast content streaming session

PPV service initiation without an existing multicast content session over UNIS-8 shall be the same as multicast content session initiation and modification as described in 7.2.3.1.1.2.

7.2.3.5.1.2 Switching from a PPV service to a multicast content service

Switching from a PPV service to a multicast content service over UNIS-8 shall be the same as multicast content streaming session initiation and modification as described in 7.2.3.1.1.2.

7.2.3.5.1.3 Switching to a PPV service from a multicast content service or another PPV

Switching to a PPV service from a multicast content service or another PPV service over UNIS-8 shall be the same as multicast content streaming session initiation and modification as described in 7.2.3.1.1.2.

7.2.3.5.1.4 Session termination

PPV multicast content streaming session termination over UNIS-8 shall be the same as Scheduled Content session termination as described in 7.2.3.1.1.3.

7.2.3.5.2 Protocol over NPI-4

7.2.3.5.2.1 PPV service initiation without existing multicast content streaming session

The IPTV control FE shall support the procedures specified in ETSI TS 183 063, 5.3.1.5.1.

The IPTV control FE shall support the procedures specified in ETSI TS 124 503 that are applicable to an AS acting as a terminating SIP UA.

Starting watching the PPV service directly shall use multicast content session initiation as described in 7.2.3.1.2.2, with the following differences:

- The IPTV control FE shall examine the SDP parameters. In particular, it shall examine the a=bc_service: parameter which contains the programme ID the OITF intends to view. The IPTV control FE shall further check this parameter with the user's profile:
 - If the OITF isn't allowed to view the programme indicated according to the user profile, the IPTV control FE shall not accept the offer and shall answer with a 403 error code.
 - If the programme is allowed in the user profile, but has not started, i.e., the current time is earlier than the programme start time, the IPTV control FE shall not accept the offer and shall answer with a 403 error code.
 - If the programme is allowed in the user profile, and has started, i.e., the current time is later than the programme start time, the IPTV control FE shall switch to the corresponding PPV service as defined in 7.2.3.5.2.3.
- The IPTV control FE shall process the a=bc_service_package attributes as defined in 7.2.3.1.2.2.

7.2.3.5.2.2 Switching from a PPV service to a multicast content service

Switching from a PPV service to a multicast content service over NPI-4 shall be the same as multicast content session modification as described in 7.2.3.1.2.3.

7.2.3.5.2.3 Switching to a PPV service from a multicast content service or another PPV

Upon receipt of a PPV session modification request, the IPTV control FE shall follow the procedures defined in ETSI ES 283 003 concerning the AS acting as a terminating UA or a B2BUA.

When receiving an SDP offer, the IPTV control FE may modify the SDP answer in accordance with the user subscription as defined in 7.2.3.5.2.1. If the IPTV control FE finds a media line not compatible with the user's subscription, it shall set the port of this media line to 0. If none of the media lines are acceptable, it shall reply with a 403 error response.

7.2.3.5.2.4 Session termination

Upon receipt of a PPV session termination request, the IPTV control FE shall follow the procedures defined in ETSI TS 124 503 concerning the AS acting as a terminating UA, which is the same as multicast content streaming session termination as described in 7.2.3.1.2.4.

Upon receipt of an internal indication that a PPV session shall be terminated, the IPTV control FE shall generate a BYE request and follow the procedures defined in ETSI TS 124 503 for an originating OIPF, which is the same as multicast content streaming session termination as described in 7.2.3.1.2.4.

7.2.3.6 Parental control for content using SIP

7.2.3.6.1 What is on TV – OITF initiated

7.2.3.6.1.1 Protocol over UNIS-8

The following shall be supported by the IG for subscription to acquire information related to content streamed at an OITF:

- An outgoing SUBSCRIBE message for subscription to the parental control Watched Content event. Note that the IPTV control FE shall reject a SUBSCRIBE message from a non-authorized user.
- An incoming NOTIFY message that complies with 6.3.7.2.4 shall be proxied to the OITF, otherwise it shall be rejected and no further processing shall be performed.
- The element Programmeld (6.3.7.2.4) may be used.
- The OITF contact information shall be mapped to DeviceId.
- The OITF Call-ID shall be mapped to sessionId.

7.2.3.6.1.2 Protocol over NPI-4

The IPTV control FE can optionally use the presence server to support the “What is on TV” feature. In that respect, the IPTV control FE acts as a publisher for the content being watched on the OITF. The usage of presence server for parental control purposes is distinguished from the usage of presence server for sharing information between users for presence purposes. The latter depends on information published directly by presence agents on behalf of users and is under user control, and reflects what the user wants to declare as far as presence information, including what he/she is currently watching. Information published by the IPTV control FE is information derived during the session establishment procedures, and is considered as state information.

If the IPTV control FE uses the presence server to provide parental control service, it shall support the SIP PUBLISH request for publishing watched content. The event in this case shall be parental control Watched Content event as per 6.3.7.2.4.

7.2.3.6.2 Parental control

7.2.3.6.2.1 Protocol over UNIS-8

When receiving a request for parental control from the OITF as described in 6.3.7.3.1, the IG shall generate an initial MESSAGE request as specified in IETF RFC 3428 IETF RFC 3428.

When receiving a request for parental control from the network as a SIP MESSAGE as described in 6.3.7.3.2, the IG shall forward this information to the OITF including the information in the body.

When the final response to the SIP MESSAGE for parental control is received from the network as a SIP 200 OK, the IG shall forward this information to the OITF. The IG shall forward the final received response from the OITF to the network.

7.2.3.6.2.2 Protocol over NPI-4

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

Upon receipt of an SIP MESSAGE request for parental control from the IG via the Authorization and session Management FE, the IPTV control FE shall identify the Content-Type associated with the MESSAGE request to determine that it is a parental control request. The IPTV control FE shall check the controlled user in the “To:” header as well as the

controller in the “From:” fields to determine whether the controller has the rights to perform parental control on the controlled user.

If the controller has the rights to perform parental control on the controlled user, the IPTV control FE shall forward the SIP MESSAGE to the controlled user.

Upon receipt of a SIP 200 OK response to the SIP MESSAGE for parental control from the OITF via the Authorization and session Management FE, the IPTV control FE shall forward the SIP 200 OK to the controller.

7.2.3.7 Network-based User Notification services using SIP

7.2.3.7.1 Protocol over UNIS-8 for Native HNI-IGI (IMS-based) notification request setup

Upon receipt by the IG for an HTTP POST to send a SIP MESSAGE, as described in 6.3.8.2 Notification request Setup Procedure”, the IG shall initiate a SIP MESSAGE which shall conform to OMA-TS-SIMPLE_IM-V1_0-20100322-C. The response to the SIP MESSAGE shall comply with OMA-TS-SIMPLE_IM-V1_0-20100322-C. The IG shall not retain any state information once the transaction is completed.

7.2.3.7.2 Protocol over NPI-4 for IMS-based notification request setup

Upon receipt by the IPTV Control Server FE of a SIP MESSAGE, the IPTV control FE shall first validate that the user is authorized to initiate the request. Following successful validation, the IPTV control FE shall issue a store request to the appropriate IPTV application and shall wait for the response.

Upon receipt of a response from the IPTV application, the IPTV control FE shall return a SIP 200 OK response to the IG, or an appropriate error response.

7.2.3.8 Content bookmarking using SIP

7.2.3.8.1 Protocol for storing content bookmarks over UNIS-8

Upon receiving a request from the OITF for storing a scheduled content bookmark or a CoD bookmark, (see 6.3.9.2.1), the IG shall generate a SIP INFO including the CoD-Bookmark Info Package according to Clause 2 of 3GPP TS 26.237.

The IG shall forward any received SIP response to the OITF.

Upon receipt of a SIP UPDATE by the IG with an empty Recv-Info header or unwillingness to receive the CoD-Bookmark Info Package from the IPTV control FE, to stop sending content bookmarks for storage, the IG shall forward the SIP UPDATE to the OITF in an HTTP 200 OK response. The IG shall wait for the response from the OITF to forward it to the IPTV control FE.

Upon receipt of a SIP UPDATE, including the Recv-Info header set to support the CoD-Bookmark Info Package, by the IG from the IPTV control FE, to start storing content bookmarks, the IG shall forward the SIP UPDATE to the OITF in an HTTP 200 OK response. The IG shall wait for the response from the OITF to forward it to the IPTV control FE.

7.2.3.8.2 Protocol for storing content bookmarks over NPI-4

Upon receipt of SIP INFO from the IG including the CoD-Bookmark Info Package according to Clause 2 of 3GPP TS 26.237 for storing a content bookmark, the IPTV control FE shall examine the body.

If the body is not successfully validated as compliant to 6.3.9.6, the IPTV control FE shall issue an appropriate error message. Otherwise, the IPTV control FE shall issue a bookmark store request to the IPTV application responsible for handling bookmarks and shall wait for the response before returning a response to the IG. The treatment of the different elements in the body is specified in 6.3.9.6.

If the IPTV control FE desires to instruct the OITF to stop sending content bookmarks for storage, it shall send a SIP UPDATE to the IG where the Recv-Info is empty or removes support for the reception of the CoD-Bookmark Info Package and shall wait for the response.

If the IPTV control FE desires to instruct the OITF to start sending content bookmark for storage, it shall send a SIP UPDATE to the IG and where the Recv-Info is set to indicate support for the reception of CoD-Bookmark Info Package and shall wait for the response.

7.2.3.8.3 Content-related bookmark retrieval

7.2.3.8.3.1 Session initiation over UNIS-8

The IG shall support the procedures specified in ETSI TS 124 503 for initiating unicast sessions.

On receiving a request for a unicast session initiation from the OITF, the IG shall generate an initial INVITE request as specified in ETSI TS 124 503 (for an originating UA). See 6.3.2.1.2.

When the final response to the SIP INVITE request is received from the network as a SIP 200 OK including the SDP (for content delivery and content control channel), the IG shall forward this information to the OITF.

When the SIP INFO request with the XML body (for content-related bookmark list) is received from the network, the IG shall forward this information to the OITF. The IG shall forward the received response from the OITF to the network.

7.2.3.8.3.2 Session initiation over NPI-4

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

When receiving the SIP INVITE request, the IPTV control FE shall send an XCAP GET request to the IPTV service profile to get the user's profile and content-related bookmark list with the user ID and content ID which are retrieved from the INVITE request and then examine whether the user has the right to initiate a unicast session for the content. If the user is not allowed to initiate a session for the requested content, the IPTV control FE shall reply with an appropriate SIP error response. If the user is allowed to initiate the session, the IPTV control FE shall forward the SIP INVITE to a default CDN controller.

The IPTV control function shall not change the user-part of the To header in order to retain the content-id in the INVITE request and maintains the SDP received from the IG.

When receiving the SIP 200 OK response for the INVITE request, the IPTV Control forwards the SIP 200 OK response to the OITF, and then sends the SIP INFO to the OITF with the XML body for the content-related bookmark list (see 6.3.9.6).

This does not apply for the case when the session is related to a user-activated time shift. In case of user-activated time shift, the IPTV control FE shall change the To field to include the requested content ID that corresponds to the shifted multicast content. When the final response is received, the IPTV control FE shall restore the TO field to its original value. In that respect, the IPTV control FE maintains this information in the session state as long as the session is alive.

7.2.3.8.3.3 Session initiation over NPI-19

The procedure is the same as that defined in 7.2.3.2.2.3.

7.2.3.8.3.4 Session initiation over NPI-25

The procedure is the same as that defined in 7.2.3.2.2.4.

7.2.3.8.4 Session initiation over NPI-26

The procedure is the same as that defined in 7.2.3.2.2.5.

7.2.3.9 Local PVR service using SIP

7.2.3.9.1 Protocol over UNIS-8

7.2.3.9.1.1 PVR service capture request/response

Upon receiving a request from the OITF for the PVR service capture request (see 6.3.10), the IG shall generate a SIP MESSAGE as specified in ETSI TS 124 503.

7.2.3.9.1.2 PVR record request/response

Upon receiving a request from the OITF for the PVR service Capture request (see subclause 6.3.10), the IG shall generate a SIP MESSAGE as specified in ETSI TS 124 503.

The IG shall forward any received SIP response to the OITF including this information.

If the IG receives PVR Record request initiated from an OITF different than the one used for recording, the IG shall forward PVR Record request to relevant OITF.

7.2.3.9.2 Protocol over NPI-4

7.2.3.9.2.1 PVR service capture request/response

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

When receiving SIP MESSAGE for PVR service capture from the IG, the IPTV control function shall perform the following:

- verifies that the user is subscribed to the service;
- verifies that there is no active capture order for the same program;
- verifies that the user is allowed to set up a scheduled recording order in the local PVR mode; when the local PVR mode is initiated, the IPTV control function verifies the recording capabilities of the target local PVR (i.e. storage of the local PVR).

The IPTV control function shall forwards an appropriate SIP response based on the outcome of the verification process.

7.2.3.9.2.2 PVR record request/response

When the IPTV control function initiates a SIP MESSAGE request for PVR recording, the request-URI shall be set to the Public Identity of the target user (IMPU). The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA. The content of the SIP MESSAGE shall conform to Table 44 as per step 5 in 6.3.10.1.

The IPTV control function shall wait for the received response.

7.2.3.10 Network PVR (nPVR) using SIP

7.2.3.10.1 OITF-initiated nPVR

7.2.3.10.1.1 Protocol over UNIS-8 PVR record request/response

Upon receiving a request from the OITF for the PVR service request (see 6.3.11.), the IG shall generate a SIP MESSAGE message as specified in ETSI TS 124 503. The IG shall forward any received SIP response to the OITF.

7.2.3.10.1.2 Protocol over NPI-4

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a terminating SIP UA.

When receiving any NPVR request, the IPTV control function shall examine the request to see:

- if the user is subscribed to the PVR service;
- if the programme is allowed to recorded;
- if it is compatible with the user's subscription profile (e.g. parental control level);
- if the new item to be recorded doesn't exceed the user's storage quota.

If the record request is successful, the IPTV control function shall create a context for the order, register relevant information and update the user's profile status for PVR to "Order Captured", meaning that a recording order is pending execution.

If the record request is not valid, the IPTV control function shall respond with a non-2XX SIP response and shall subsequently issue a SIP MESSAGE compliant to the OITF to report the reason for rejection.

The body of the SIP MESSAGE shall include the MIME type "application/vnd.oipf.pvrresult+xml" based upon the XML schema defined in 6.3.11.2.

At the start of the scheduled programme or the start time of the immediate NPVR request, the IPTV control function shall find the appropriate CDNC/CC/CDF to setup the content delivery channel for the content recording. It is done by issuing an SIP INVITE to the selected Content Delivery Network control function, and shall wait for 200 OK response.

Based on the local policy, the following should apply to avoid duplicated recording:

- Upon receiving a scheduled NPVR record request, the IPTV control FE checks whether the same programme (identified by BCserviceID and ProgramID) has been requested to be recorded. If yes, the IPTV control FE should not issue another NPVR request for the same content.
- Upon receiving an immediate NPVR record request, the IPTV control FE examines the status of the program. If the same programme (identified by BCserviceID and ProgramID) is already under recording, the IPTV control FE should not issue another NPVR request for the same content.

NOTE In case of duplicated recording for multiple users, the users will have the same CRID for accessing the recorded content.

The content of the SIP INVITE message shall be as follows:

For the header, see Table 132.

The body: the message body shall include a Multipart/Related MIME container conforming to IETF RFC 2387, containing an SDP of the multicast content to be recorded and a nPVR

request XML document, with the “requestType” set to “Setup”. The nPVR request XML document shall conform to 6.3.11.2.

Table 132 –SIP headers for PVR service recording status (CC→IPTV Control)

SIP header	Source of coding information
request-Line Note that the request URI shall be set to the SIP URI of the selected CDNC	IETF RFC 3261 INVITE <request URI> SIP/2.0
From	IETF RFC 3261
To The URI part of To shall be set to the value of the request URI in the “request-Line”	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Content-Type It shall be set to “multipart/related”	IETF RFC 3261
Content-Length	IETF RFC 3261

Upon receiving 200 OK response from the CDNC, the IPTV control FE shall update status of the NPVR request in the users’ profile with “Recording”.

Upon receiving a SIP UPDATE message from the CDNC, the IPTV control FE shall extract the programme identifier and the status information, then update the user profiles of all the users who requested to record the programme in the network. Then the IPTV control FE shall generate a BYE request as specified in ETSI TS 124 503 for originating sessions.

7.2.3.10.1.3 Protocol over NPI-19

The CDN controller FE shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP proxy or B2BUA.

When receiving the SIP INVITE from the IPTV control FE via the authentication and session Management FE through the NPI-19 reference point, the CDN controller shall check the request type and the programme id of the NPVR request, and determine the most appropriate cluster controller FE to serve the NPVR request.

Once the appropriate cluster controller FE is selected, the Content Delivery Network Controller FE shall forward the SIP INVITE to it by changing the “request-URI” accordingly.

The CDN controller shall not forward 301 or 302 responses from the cluster controller to the IPTV control function. The CDN controller shall take one of the following actions on receiving a 301 or 302 response from the cluster controller:

- cancel the transaction;
- forward to another cluster controller;
- forward to the suggested CC as indicated in the 301/302 response;
- forward to another CDN controller.

Upon receiving a SIP UPDATE message from the cluster controller, the CDNC shall forward the message to the IPTV control FE.

7.2.3.10.1.4 Protocol over NPI-25

On receiving the request from the IPTV control function, the CDN controller may decide to forward the request to another CDN controller. In this case it changes the “request-URI” accordingly.

7.2.3.10.1.5 Protocol over NPI-26

The cluster controller FE shall support the procedures specified in ETSI TS 124 503 as applicable to a terminating UA.

Upon receiving a NPVR request from the CDN controller, the cluster controller shall examine the programme identifier present in NPVR request XML document, and the media parameters in the received SDP offer and then choose a CDF.

If all the CDFs lack of storage, the cluster controller shall return a 301 response, or a 302 response for any other reasons (e.g. load-balancing). The cluster controller may indicate one or more cluster controller addresses in the contact header as indicated in IETF RFC 3261.

If the request is not acceptable to the cluster controller, it shall reply with an appropriate SIP error response.

The cluster controller shall reply with an appropriate SIP error response if the request is acceptable to the cluster controller but none of the content delivery functions can handle the offer.

If the request is acceptable to the cluster controller and a CDF can handle the request, the cluster controller shall initiate an RTSP session using the RTSP SETUP message to the chosen CDF. The CDF then shall join the multicast group, and return 200 OK with the server port and RTSP session ID to setup the content delivery channel between the CDF and the content source. Then CC sends a RTSP Recording message to the CDF to record the content as defined in 8.2.3.1.

Following the successful conclusion of the RTSP session setup, the cluster controller allocates an RTSP server port, binds it to the CDF RTSP server port and answers with a SIP 200 OK, including the SDP answer and the record status report conforming to 6.3.11.2.

When completing the record of the requested program, the cluster controller shall issue a SIP UPDATE message to the IPTV control FE with appropriate record status (i.e. “Recording Completed”).

The content of the above two messages shall be as follows:

- The SIP UPDATE message header: see Table 133.
- The SIP UPDATE message body: The cluster controller shall include a body associated with the appid “urn:oipf:service:PVR:2011” as defined in 6.3.11.2.

Table 133 –SIP headers for PVR service recording status (CC→IPTV Control)

SIP header	Source of coding information
request-Line Note that the request URI shall be set to well-known PSI for NPVR	IETF RFC 3261 UPDATE <request URI> SIP/2.0
From	IETF RFC 3261
To The URI part of To shall be set to the value of the request URI in the “request-Line”	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Content-Type It shall be set to “application/vnd.oipf.pvr+xml”	IETF RFC 3261
Content-Length	IETF RFC 3261

7.2.3.10.2 Non-OITF-initiated nPVR

7.2.3.10.2.1 General

The nPVR service can also be initiated by non-OITF based user equipment so that the IPTV user can consume the multicast content later with an OITF. In this case, the user will send an nPVR request to the IPTV application (or other specific FE in the service provider domain) by means such as web browser through internet / SMS through mobile network etc.

Upon receiving such nPVR request, the IPTV application shall validate the user and then, on behalf of the user, send the nPVR request to the IPTV control FE. The IPTV control FE shall validate the request, do recording and update the user profile, as defined in 7.2.3.10.1.

7.2.3.10.2.2 Protocol over NPI-19

The protocol used in NPI-19 for non-OITF initiated the nPVR service shall be the same as for OITF initiated network PVR service, as defined in 7.2.3.10.1.3.

7.2.3.10.2.3 Protocol over NPI-25

The protocol used in NPI-25 for non-OITF initiated nPVR service shall be the same as for OITF initiated network PVR service, as defined in 7.2.3.10.1.4.

7.2.3.10.2.4 Protocol over NPI-26

The protocol used in NPI-26 for non-OITF initiated nPVR service shall be the same as for OITF initiated network PVR service, as defined in 7.2.3.10.1.5.

7.2.3.11 Personalised channel using SIP

7.2.3.11.1 Procedure for network-centric personalised channel (unicast only)

7.2.3.11.1.1 Protocol over UNIS-8, NPI-19, NPI-25 and NPI-26

The procedures over UNIS-8, NPI-19, NPI-25 and NPI-26 for PCh service shall follow those described in 7.2.3.2, with the difference that the request URI and the To header in the INVITE request from the OITF shall include the indicated personalised channel identifier.

7.2.3.11.1.2 Protocol over NPI-4

7.2.3.11.1.2.1 PCh session initiation

The IPTV control function shall support the procedures specified in ETSI TS 124 503 as applicable to an AS acting as a SIP B2BUA.

When receiving any SIP request, the IPTV control FE shall examine the request to see if it is compatible with the user's subscription profile (e.g. parental control level). If the user is not allowed to initiate a session for the requested personalised channel, the IPTV control FE shall reply with an appropriate SIP error response.

If the user is allowed to initiate the session, the IPTV control FE shall first retrieve the detailed PCh information for the user from the IPTV application FE via NPI-2 reference point, then it shall forward the SIP INVITE to a default CDN controller with the following modifications:

- The request URI shall be set to the public identity of the CDN controller FE;
- The To header shall include the PChId and the content item which is to be played, i.e. PChId: BCserviceld or PChId: COD content Id, which are extracted from the PCh information.

The request body includes the SDP identical to the incoming INVITE request, i.e. it shall contain the unicast content delivery description for the OITF regardless whether the next PCh item is scheduled content or on-demand content.

7.2.3.11.1.2.2 PCh content switch

When it is time for the next PCh content to be played, the IPTV control FE shall initiate SIP INFO based upon the configuration of the retrieved PCh information. The IPTV control FE shall construct the out-going SIP INFO request to the selected CDNC/CC/CDF, as described below:

- request URI shall include the PChId and the content item which is to be played, i.e. PChId: BCserviceld or PChId: COD content Id, which are extracted from the PCh information;
- the To header shall be the same with the request URI;
- the From header shall include the public identity of the IPTV control or the well-known PSI of PCh service;
- the Content-Type header shall be set to application/vnd.oipf.pchcontentswitch+xml, conforming to IETF RFC 3261;
- the message body shall include one or more "PChContentSwitchControl" elements.

NOTE The detailed XML schema refers to 7.2.3.11.1.3.

When the CC receives the SIP MESSAGE, it shall respond a SIP 200 OK, and then parse the message body for content switch, i.e. use the ContentId and SwitchTime to initiate the RTSP PLAY towards the CDF for delivery of the next PCh content item as described in 8.2.4.2.

7.2.3.11.1.2.3 PCh session termination

Session termination for PCh shall follow the same SIP procedures as session termination for the COD service. See 7.2.3.2.3.

7.2.3.11.1.3 XML schema for PCh content switch

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:pchcontentswitch:2011"
  xmlns:tns="urn:oipf:iptv:pchcontentswitch:2011"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd" />
  <xs:element name="PChContentSwitchControl">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PChContentItem" type="tPChContentItem"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="PChId" type="xs:anyURI" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="tPChContentItem">
    <xs:sequence>
      <xs:element name="ProgramIdentifier" type="ct:ProgramIdType" />
      <xs:element name="SwitchTime" type="xs:dateTime" />
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

7.2.3.11.2 OITF-centric personalised channel protocol over UNIS-8, NPI-4, NPI-19, NPI-25 and NPI-26

When recording overlapped content, the local PVR or nPVR procedures will be used.

The messaging and procedures for recording the overlapped content on a local PVR shall be as specified in 7.2.3.8.3.

The messaging and procedures for recording the overlapped content on a nPVR shall be as specified in 7.2.3.10.

When a time gap between adjacent content items is detected, the OITF may perform the procedures specified in 6.3.12.2.

7.2.3.12 Session transfer with SIP session management

7.2.3.12.1 Protocol over UNIS-8

7.2.3.12.1.1 Generic procedures

7.2.3.12.1.1.1 Session initiation procedures related to a session transfer

The IG shall support the procedures specified in 3GPP TS 24.237 for initiating unicast sessions associated with session transfer.

On receiving a request for a unicast session initiation from the transferee OITF, the IG shall generate an initial INVITE request as specified in 3GPP TS 24.237.

7.2.3.12.1.1.2 IG initiated session modification procedures

The IG shall support the procedures specified in ETSI TS 124 503 for putting media subject to transfer, on hold (setting the port to 0 in the m-line in the SDP) and reducing the QoS resources for the transferor OITF down to zero, when it detects that the transferor and the transferee are behind the same IG.

7.2.3.12.1.2 Session transfer via push mode

The following shall be supported by the IG:

- an outgoing REFER message associated with transferor OITF shall conform to 3GPP TS 24.237;
- the IG shall forward any incoming SIP REFER, for the purpose of session transfer, that conforms to the 3GPP TS 24.237 to the transferee OITF; non-conformant SIP REFER messages shall be rejected with the appropriate response code;
- an outgoing SIP NOTIFY message associated with the transferee reporting the outcome of a session transfer associated with a REFER request, shall conform to 3GPP TS 24.237 and IETF RFC 3515;
- the IG shall forward any incoming SIP NOTIFY, reporting the outcome of a session transfer associated with a REFER request, that conforms to 3GPP TS 24.237 and IETF RFC 3515 to the transferor OITF; non-conformant SIP REFER messages shall be rejected with the appropriate response code.

7.2.3.12.2 Protocol over NPI-4

7.2.3.12.2.1 Generic procedures

7.2.3.12.2.1.1 Session initiation procedures

When receiving a SIP INVITE request associated with a session transfer, the IPTV control FE shall first authorize the request. If the user is not authorized or the INVITE is not successfully validated, in accordance with 3GPP TS 24.237 the IPTV control FE shall respond with an appropriate SIP error message. If the user is authorized and the INVITE is successfully validated, then the IPTV control FE behaviour depends on the SDP in the incoming INVITE.

- If the same CDF of the transferred session can be used, then the IPTV control FE shall send a SIP UPDATE (or re-INVITE) to the selected CNDC to update the SIP session leg with the new SDP.
- If the same CDF of the transferred session cannot be used, and a new one is required, then the IPTV control FE shall terminate the SIP leg towards the old CDNC/CC/CDF to terminate the content flow sent to the transferor device first. The IPTV control FE shall create a new SIP session leg by forwarding the SIP INVITE to a default CDNC as per 7.2.3.1.1.2.

7.2.3.12.2.1.2 IG initiated session modification procedures

When receiving a SIP UPDATE (or re-INVITE) request associated with a session transfer, and related to an ongoing session, the IPTV control FE shall first validate the request. If the SIP UPDATE (or re-INVITE) is successfully validated, then the IPTV control FE behaviour shall release the resources based on the incoming SIP UPDATE (or re-INVITE), shall put the media on hold, and shall return the received response to the IG.

7.2.3.12.2.2 Session transfer via push model

When receiving a SIP REFER request, the IPTV control FE shall authorize the request. If the user is not authorized to perform session transfer, an appropriate SIP error response is returned. If the user is authorized, the IPTV control FE shall forward the SIP REFER to the ASM for delivery to its destination (transferee OITF). The IPTV control FE shall be stateful to the session transfer procedure

When receiving a SIP NOTIFY associated with a SIP REFER request, the IPTV control FE shall validate the SIP NOTIFY as per 3GPP TS 24.237. If not successfully validated, an appropriate SIP response shall be returned. If successfully validated, the IPTV control FE shall forward the SIP NOTIFY to the ASM for delivery to the destination (transferor OITF).

7.2.4 Protocols for service access and control functions

7.2.4.1 Service provider discovery - Protocol over UNIS-8 and NPI-30

The IPTV service provider discovery FE shall generate and/or provide the service provider discovery information.

The IG shall follow the following procedure to retrieve service provider discovery information:

- Step 1:** The IG shall send a SIP SUBSCRIBE to the network, to subscribe to the “ua-profile” event, and shall wait for the response to the subscription request.
- Step 2:** When a SIP NOTIFY is received by the IG for a “ua-profile” event, the IG shall store the body of the SIP NOTIFY.
- Step 3a:** If the IG receives a HTTP GET for the service provider information, it shall return the body of the SIP NOTIFY (from step 2) in the HTTP response body.
- Step 3b:** If the IG receives a HTTP POST on the HNI-IGI interface from the OITF which includes a SIP SUBSCRIBE with a message body associated with the appid “urn:oipf:application:iptv-SP-discovery”, the IG shall send a SUBSCRIBE to the network with the following capabilities.

The message body shall include what was received from the OITF, which are the capabilities of the OITF which are sent to the service provider discovery FE. The details of the SIP SUBSCRIBE are as specified in ETSI TS 183 063, 5.1.2.2.1, namely:

- the Content Type header shall be set to “application/vnd.oipf.ueprofile+xml”;
- the UserEquipmentID is a unique global identifier of the device;
- the User Equipment Class shall take either of the values “TV-OITF”, “STB-OITF”, as appropriate.

When the service provider discovery FE receives a SUBSCRIBE request, it may check the user’s IPTV subscription profile and provide a personalized service provider discovery information to the OITF. Filtering may also be performed if device capabilities are available to the SDF.

If the service provider discovery FE receives a SIP SUBSCRIBE message body from the IG carrying OIPF capabilities, the service provider discovery FE shall process the SIP request as specified below.

On successful subscription, the service provider discovery FE shall generate a 200 OK response. The service provider discovery FE shall then send a NOTIFY request to the OITF in accordance with IETF RFC 3265.

The contents of the SIP NOTIFY request shall be as follows:

- Extend the existing “ua-profile” event package for SIP NOTIFY as follows:
 - the Event header shall be set to the “ua-profile” event package;
 - the “effective-by” parameter for the event header shall be set to 0;
 - the content type shall be set to “application/vnd.oipf.spdiscovery+xml”.

The service provider discovery information shall be delivered in the message body and shall conform to the schema defined in IEC 62766-3.

Note that if the above extension is not accepted by the IETF, then the use of a new method (New Event package) should be re-examined (see Annex L).

When the IPTV service provider discovery FE knows of a change to the service discovery, service provider or Selection Information, the IPTV service provider discovery FE shall inform the OITF of this change by sending a SIP NOTIFY message.

7.2.4.2 User registration and network authentication

7.2.4.2.1 User identity modelling

Every IMS Subscription shall be allocated a single unique default IMS Public Identity by the service platform provider. This shall be the identity that is registered in the IMS domain when an OITF is turned on.

Every IPTV end-user in an IMS Subscription may be associated with an IMS Public User Identity by the service platform provider.

7.2.4.2.2 Procedure for user registration and authentication in a network relying on IMS on UNIS-8

The following shall be supported by the IG on the UNIS-8 interface for user registration:

- Upon receipt of a registration request from the OITF, the IG shall maintain a binding between the sip instance feature tags and other feature tags declared in the registration request against the registered user, and the OITF device.
 - Furthermore if the incoming registration request from the OITF includes a request for a GRUU, the IG shall allocate a new user name to the username (IMPU) portion of the URI (username@host) in the contact header information (see Annex O why the IG needs to do that). The new username will replace the existing username (IMPU) in the URI contact header information before the IG registers the user with the network.
 - The IG shall maintain a binding between the IMPU included by the OITF in the incoming HNI-IGI request, the new username created by the IG, and the actual OITF device (extracted from the sip instance feature tag) from which the request came.
- The IG shall support the 3GPP IMS registration procedure as per ETSI TS 124 503. This includes handling of user authentication and authorization. This procedure shall be invoked when the IG powers up (in this case the default identity shall be registered) or upon receipt of an HTTP POST from the OITF with the REGISTER method.
- The IG shall report to the OITF the final outcome of any OITF-initiated registration or de-registration.
- The IG shall be stateful for all successful registrations until de-registration occurs.
- For IG-initiated registration procedures, the IG is responsible for refreshing the registration before the registration expiry time.

The following shall be supported by the IG on the UNIS-8 interface for user de-registration:

- The IG shall support the 3GPP IMS de-registration procedure as per ETSI TS 124 503. This procedure shall be invoked upon receipt of an HTTP POST from the OITF with the REGISTER method and when the IG shuts down. Following the successful de-registration process, the IG shall remove all bindings related to the de-registered IMPU.

The following shall be supported in the IG on UNIS-8 for subscription to the Registration event:

- For OITF-initiated registrations, the IG shall support subscription to the registration-state event package as per ETSI TS 124 503.
- For IG-initiated registrations, and following a successful registration process, the IG shall SUBSCRIBE to the registration event package in accordance with ETSI TS 124 503.

- The IG shall always validate the XML schema associated with the registration event package for both OITF and IG initiated registrations.
- On request from the OITF, as well as for IG-initiated registrations, the IG shall refresh the registration-state event package subscription in accordance with ETSI TS 124 503.
- For OITF-initiated registrations, the IG shall not store any registration event related data, but shall be stateful of the subscription. For IG-initiated registrations, the IG shall store registration event related data.
- For OITF-initiated registrations, the IG shall support terminating a subscription to the registration-state event package as per ETSI TS 124 503.
- For IG-initiated deregistration, and following the successful deregistration process, the IG shall terminate the subscription to the registration event package, as per ETSI TS 124 503.

The appropriate procedure (SIP Digest or IMS AKA) shall be followed by the IG for user registration and authentication.

7.2.4.3 Notification of service profile changes protocol over UNIS-8

7.2.4.3.1 Subscription to notifications of service profile changes

If subscription to notification of changes is requested by the OITF, the IG shall send a SUBSCRIBE request to the IPTV service profile FE in accordance with IETF RFC 5875 and IETF RFC 5874.

The IG will process the request from the OITF and will generate a SUBSCRIBE request, that shall be as specified in ETSI TS 183 063, 5.1.5.1.

A well-known PSI mechanism shall be used in the request URI of the SUBSCRIBE request.

NOTE For changes that apply to a very large number of subscribers, it is up to service provider to set up proper rules in the 'notifier function' to make the notification procedure scalable.

7.2.4.3.2 Processing of notifications

Refer to ETSI TS 183 063, 5.1.5.2.

7.2.5 Protocols for communication services using SIP CallerID Procedures for caller ID on UNIS-8

7.2.5.1 Instant message based caller ID

Instant message based Caller ID is identical to instant messaging where the incoming message includes the caller ID. For further details reference should be made to 7.2.5.3.

7.2.5.2 IMS telephony service based caller identification

IMS telephony service based caller identification is based on the reception of the regular SIP session INVITE request. The incoming session request message includes the caller identification.

Support of this feature by the IG is optional.

7.2.5.3 Instant messaging Procedure for instant messaging on UNIS-8

Instant messaging complies with the page mode of operation. The following shall be supported by the IG:

- incoming SIP MESSAGE messages to the IG shall conform to OMA Instant Messaging using SIMPLE Draft OMA-TS-SIMPLE_IM-V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-

20100322-C to be acceptable for processing by the IG. Non-conformant SIP MESSAGE messages shall be rejected in accordance with OMA-TS-SIMPLE_IM-V1_0-20100322-C with the appropriate response code.

- For an incoming SIP MESSAGE to the IG that is conformant to OMA-TS-SIMPLE_IM-V1_0-20100322-C, the IG shall forward the message to the OITF using HNI-IGI Notification procedure and shall send a 202 Accepted to the originating end.
- An outgoing SIP MESSAGE shall conform to OMA-TS-SIMPLE_IM-V1_0-20100322-C. The response to the SIP MESSAGE shall comply with OMA-TS-SIMPLE_IM-V1_0-20100322-C.
- The IG shall not retain any state information once the transaction is completed.

7.2.5.4 Presence

7.2.5.4.1 Procedure for presence on UNIS-8

The following shall be supported by the IG for subscription to presence:

- An outgoing SUBSCRIBE message for a subscription to the presence event, or cancellation of an existing subscription shall comply with OMA-ERP-Presence_SIMPLE-V1_1-20080627-A.
- An incoming NOTIFY messages that does not comply with OMA-ERP-Presence_SIMPLE-V1_1-20080627-A shall be rejected with the appropriate error code in accordance with OMA-ERP-Presence_SIMPLE-V1_1-20080627-A, and no further processing shall be performed.
- On request from the OITF, the IG shall refresh the Presence subscription in accordance with OMA-ERP-Presence_SIMPLE-V1_1-20080627-A.
- The IG shall not store any presence related data, but shall be stateful to the subscription.
- The IG shall consider a subscription terminated if it is not renewed by the OITF.

7.2.5.4.2 Procedure for publishing presence information on UNIS-8

When requested by the OITF, the IG shall support the SIP PUBLISH request and response in accordance with OMA-ERP-Presence_SIMPLE-V1_1-20080627-A for publishing presence information.

7.2.5.5 Chatting

7.2.5.5.1 IM session using MSRP over UNIS-8

The IG shall conform to the Client Procedure as described in OMA-TS-SIMPLE_IM-V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C.

The IG shall perform path mapping between chatting peers as indicated in 6.5.3.

The IG shall handle translation of chat session initiation and teardown procedures when requested by the OITF as per 6.4.5.

The IG shall handle the translation of outgoing and incoming MSRP chat messages as per 6.5.3.

7.2.5.5.2 IM session using MSRP over NPI-3

The P2P chatting communication enablers FE shall conform to the IM Server procedures described in OMA-TS-SIMPLE_IM-V1_0-20080820-D OMA-TS-SIMPLE_IM-V1_0-20100322-C.

7.2.5.6 Content sharing Procedures for content sharing on UNIS–8

Incoming SIP OPTIONS be acceptable for processing by the IG. Non-conformant SIP OPTIONS shall be rejected in with the appropriate response code.

When requested by the OITF, the IG shall support the SIP OPTIONS request and response in accordance with 6.5.5.1 for capability querying.

Incoming SIP INVITE shall be acceptable for processing by the IG. Non-conformant SIP INVITE shall be rejected in with the appropriate response code.

Content sharing is based on the reception of the regular SIP session INVITE request. The incoming session request message includes the SDP body.

Incoming SIP REFER shall be acceptable for processing by the IG. Non-conformant SIP REFER shall be rejected in with the appropriate response code.

Incoming SIP UPDATE shall be acceptable for processing by the IG. Non-conformant SIP UPDATE shall be rejected in with the appropriate response code.

7.3 SIP/SDP reference points within the residential network

7.3.1 General

This subclause defines the protocol for the use of SIP and SIP/SDP within the residential LAN over the HNI-IGI interface.

This is an alternative option to the HTTP-based HNI-IGI option specified in Clause 6. As such, the functionality is equivalent. Hence the SIP headers used in the SIP-based HNI-IGI option are identical to the ones used in the HTTP-based option (prefixed with X-OITF) in addition to the headers needed for the SIP state machine and which obviously were not needed in the HTTP-based HNI-IGI option.

The description of this option shall always reference the various tables in Clause 6 when referencing SIP headers. The referenced SIP headers in all these cases are the SIP headers encoded as HTTP headers and where the prefix “X-OITF” is included the SIP header name.

7.3.2 IG as a B2BUA

The IG acts as a transparent B2BUA. The following defines the behaviour of the B2BUA of the IG:

- The IG shall handle all IMS specific SIP headers; storing what is received from the network for re-insertion during outgoing requests (e.g. P-Preferred-Identity) according to ETSI TS 124 503, and stripping those headers, before a request/response is sent to the OITF.
- The IG, being service aware, shall support the algorithm defined in Annex Q for all OITF-initiated sessions. The IG shall insert the appropriate SIP headers matching the requested service from the OITF conforming to the appropriate service specification in that regard. For undefined services, any additional information, if required, shall be configured in the IG. The IG shall be service aware as well for all incoming sessions from the network. services supported by the IG shall be identical to the ones supported for OITF-initiated sessions
- The IG shall behave transparently to all SIP headers defined in IETF RFC 3261, that are received from the network, and shall not alter them before being sent to the OITF. The same applies to SIP headers sent from the OIT to the network. The only exception being

Call-ID, and where the IG shall insert its own, and shall bind it to the Call-ID of the OITF for the duration of the session.

- The IG shall validate SIP requests/responses before accepting them. The various services specify the validation to perform by the IG.
- The IG Shall only validate compliance to XML schemas associated with event packages.
- The IG shall validate SDP syntax for correctness.
- In regard IMS registration, the IG shall perform the IMS registration when requested by the OITF. The IG is therefore stateful to IMS registration. However, graceful de-registration and re-registration shall be triggered by the OITF. The IG shall deal with all non-graceful circumstances.
- The specific subclauses on IMS registration specify detailed behaviour for the OITF and IG.
- The IG shall be stateful, to all IMS sessions. Session initiation, termination, and session refresh can be triggered by the OITF or the network, depending on the specific service. The IG shall deal with non-graceful circumstances.

7.3.3 Protocols for IPTV service functions

7.3.3.1 Multicast content streaming with SIP session management

7.3.3.1.1 General

The IG shall support the procedures specified in ETSI TS 124 503 for originating sessions.

7.3.3.1.2 Session Initiation

To initiate a multicast content streaming session, the OITF shall initiate a SIP INVITE to the IG that includes all the SIP headers in Table 5 and any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The OITF shall also include an SDP offer conformant to Step 1 in 6.3.1.2.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INVITE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 5. Following a successful validation, the IG shall generate an initial SIP INVITE request as specified in ETSI TS 124 503 for originating sessions.

The IG shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 6 and all mandatory headers as per IETF RFC 3261.

When the OITF receives the response to the SIP INVITE, it shall examine the media parameters in the received SDP. The OITF shall restrict the multicast content services that it joins according to the parameter (the a=bc_service_package attribute) received from the IPTV control FE. However, if the OITF retrieved the IPTV user profile prior to session initiation, then it may ignore the =bc_service_package attribute.

If the OITF receives an error code with an Insufficient Bandwidth indication in the response from the IG, the OITF may perform a new SIP INVITE with a reduced maximum bandwidth for the multicast content service. This procedure may be repeated. If no agreement can be reached, the OITF may display a failure message to the user.

Finally to complete the SIP INVITE transaction, the OITF shall send a SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 7 and all mandatory headers as per IETF RFC 3261.

7.3.3.1.3 Session modification

To join a service outside the set of channels negotiated at session initiation, or to perform a bandwidth modification, the OITF shall generate a SIP re-INVITE request that includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 5.

The OITF shall include an SDP offer in the session modification request. The format of this request shall be the same as for a session initiation.

The IG shall handle session modification similar to session initiation.

7.3.3.1.4 Session termination

To terminate a session, the OITF shall send a SIP BYE request to the IG that includes all headers listed in Table 8 and any other mandatory headers as per IETF RFC 3261.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP BYE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 8. Following a successful validation, the IG shall generate a SIP BYE as specified in ETSI TS 124 503 for originating sessions. The IG shall forward the received response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 9 and all mandatory headers as per IETF RFC 3261.

Alternatively, on receipt of a SIP BYE request from the IPTV control FE, the IG, after validating the request, shall forward the request to the OITF. The OITF shall respond with a SIP 200 OK response, which the IG forwards to the IPTV control FE.

7.3.3.1.5 Session refresh

It is the responsibility of the OITF to refresh the multicast content streaming session, as per IETF RFC 3556 before the session expires. The IG shall consider a session terminated if it is not refreshed.

7.3.3.1.6 Content reporting by an OIPF

To report watched content, the OITF shall send a SIP INFO request to the IG based on the SIP INFO framework, the SIP INFO request shall include all headers listed in Table 10 and all mandatory headers as per IETF RFC 3261.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INFO includes all mandatory headers as per IETF RFC 3261 and the headers listed in Table 10. Following a successful validation, the IG shall generate a SIP INFO as specified in ETSI TS 124 503 for originating sessions. The IG shall forward the received response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 11 and all mandatory headers as per IETF RFC 3261.

7.3.3.1.7 Management of content reporting

At any time, the IG can receive an SIP UPDATE to order the OITF to stop or start the reporting of watched multicast content.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP UPDATE includes all mandatory headers as per IETF RFC 3261 headers listed in Table 12. Following a successful validation, the IG shall forward the SIP UPDATE to the OITF. The IG shall forward the received response to the network. An SIP 200 OK response shall include the SIP headers defined in Table 13 and all mandatory headers as per IETF RFC 3261.

7.3.3.1.8 User-initiated activation of network-based multicast content streaming time shift

To activate time shift for a watched multicast content service, the OITF shall initiate an SIP re-INVITE to the IG that includes all the SIP headers in Table 5 and any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The OITF shall also include an SDP offer conformant to Step 1 in 6.3.1.8.1.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INVITE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 5. Following a successful validation, the IG shall generate an initial SIP INVITE request as specified in ETSI TS 124 503 for originating sessions.

The IG shall forward any received SIP response to the OITF including the information in the SDP. An SIP 200 OK response shall include the SIP headers defined in Table 6 and all mandatory headers as per IETF RFC 3261.

When the OITF receives the response to the SIP re-INVITE, it shall examine the media parameters in the received SDP. The OITF shall conform to step 3 in 6.3.1.8.1. The OITF shall store the parameters a:fmp:iptv rtsp h-session, a:fmp:iptv rtsp h-offset, and a:fmp:iptv rtsp h-uri for later usage.

Finally, to complete the SIP INVITE transaction, the OITF shall send an SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 7 and all mandatory headers as per IETF RFC 3261.

7.3.3.1.9 User-initiated deactivation of network-based multicast content streaming time shift

To deactivate a time shift for a watched multicast content service, the OITF shall initiate an SIP re-INVITE to the IG that includes all the SIP headers in Table 5 and any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The OITF shall also include an SDP offer conformant to Step 1 in 6.3.1.8.2.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INVITE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 5. Following a successful validation, the IG shall generate an initial SIP INVITE request as specified in ETSI TS 124 503 for originating sessions.

The IG shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 6 and all mandatory headers as per IETF RFC 3261.

Finally, to complete the SIP INVITE transaction, the OITF shall send a SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 7 and all mandatory headers as per IETF RFC 3261.

7.3.3.1.10 Retrieval of bandwidth parameter for FCC and/or RET enabled multicast content service

If the OITF intends to use FCC and/or RET, when the multicast content service is FCC and/or RET enabled, the OITF shall use the procedure defined in 6.3.2.1.1 with the following modifications:

- The request-URI in the method line is set to well-known PSI for the multicast content service.
- The OITF shall include an HTTP body that shall include an SDP that includes one m-line that matches the m-line associated with the multicast content service the OITF intends to join first (see 6.3.1.2) with the following exceptions:

- no b-line is included;
- the following additional a attribute shall be included when the OITF intends to use RET:
 - a=rtcp-fb:<fmt> nack

where <fmt> indicates the RTP payload type of the IP multicast stream that carries the content service;
- the following additional a attribute shall be included when the OITF intends to use FCC:
 - a=rtcp-fb:<fmt> nack rai

where <fmt> indicates the RTP payload type of the IP multicast stream that carries the content service.

The returned response shall include total bandwidth of the multicast content service with the highest bandwidth and the overhead bandwidth required for FCC and/or RET.

7.3.3.2 Unicast content streaming with SIP session management

7.3.3.2.1 Retrieving missing parameters in the SDP prior to session setup using SIP OPTIONS

If the OITF does not have all the necessary parameters, namely the FEC information including bandwidth for FEC streams and the transport protocol, to form the SDP for a unicast content streaming session, the OITF shall retrieve missing SDP parameters using the following procedure.

The OITF shall initiate a SIP OPTIONS request to the IG that includes all the SIP headers in Table 14 and any other mandatory headers as per IETF RFC 3261.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP OPTIONS includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 14. Following a successful validation, the IG shall generate an initial SIP OPTIONS request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 15 and all mandatory headers as per IETF RFC 3261. The SDP included in the SIP 200 OK response shall include the missing parameters according to 7.2.3.2.1.2.

7.3.3.2.2 Session initiation

To initiate a unicast content streaming session, the OITF shall initiate a SIP INVITE to the IG that includes all the SIP headers in Table 16 and any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The OITF shall also include an SDP offer conformant to Step 1 in 6.3.2.1.2.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INVITE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 16. Following a successful validation, the IG shall generate an initial SIP INVITE request as specified in ETSI TS 124 503 for originating sessions.

The IG shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 17 and all mandatory headers as per IETF RFC 3261. For a description of the received SDP answer, refer to 6.3.2.1.2.

When parsing the b=RR:<bandwidth-value> line in the received SDP answer: if the bandwidth value agreed is non-zero, then the OITF shall send RTCP RRs and shall not send RTSP keep-alive messages. If the bandwidth value received is zero, then the OITF shall not send RTCP RRs but instead it shall send RTSP keep-alive messages.

If the OITF receives an error code with an Insufficient Bandwidth indication in the response from the IG, the OITF may perform a new SIP INVITE with a reduced maximum bandwidth for the content. This procedure may be repeated. If no agreement can be reached, the OITF may display a failure message to the user.

Finally to complete the SIP INVITE transaction, the OITF shall send a SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 18 and all mandatory headers as per IETF RFC 3261.

7.3.3.2.3 Session termination

To terminate a session, the OITF shall send a SIP BYE request to the IG that includes all headers listed in Table 19 and any other mandatory headers as per IETF RFC 3261.

The IG, acting as a B2BUA in accordance with subclause 7.3.2 shall validate that the SIP BYE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 19. Following a successful validation, the IG shall generate a SIP BYE as specified in ETSI TS 124 503 for originating sessions. The IG shall forward the received response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 20 and all mandatory headers as per IETF RFC 3261.

Alternatively, on receipt of a SIP BYE request from the IPTV control FE, the IG, after validating the request, shall forward the request to the OITF. The OITF shall respond with a SIP 200 OK response, which the IG forwards to the IPTV control FE.

7.3.3.2.4 Session refresh

It is the responsibility of the OITF to refresh the unicast content streaming session, as per IETF RFC 3556 before the session expires. The IG shall consider a session terminated if it is not refreshed.

7.3.3.3 Pay per view multicast content service with SIP session management

7.3.3.3.1 PPV service initiation without existing multicast content streaming session

The OITF shall follow the same procedure for multicast content service in 7.3.3.1.2, to initiate a PPV session. The SDP shall conform to 7.3.3.1.2, and 6.3.6.2.

7.3.3.3.2 Switching from a PPV service to a multicast content service

To join a multicast content service outside the set of negotiated channels for the ongoing scheduled session, the OITF shall request session modification as per 7.3.3.1.3.

If the channel the OITF intends to join is already negotiated, the OITF shall follow the normal procedures for leaving and joining multicast channels.

7.3.3.3.3 Switching to a PPV service from a multicast content service or another PPV

If the PPV service the OITF intends to join is outside the set of channels negotiated within the ongoing scheduled session, the OITF shall request session modification as per 7.3.3.1.3.

If the channel the OITF intends to join is already negotiated, the OITF shall follow the normal procedures for leaving and joining multicast channels.

7.3.3.3.4 Session termination

Session termination is identical to session termination of a multicast content service as depicted in 7.2.3.1.1.3.

7.3.3.4 Parental control for scheduled content using SIP

7.3.3.4.1 What is on TV – OITF initiated

7.3.3.4.1.1 Procedure for subscription to parental control watched content event package (OITF-initiated)

To subscribe to the event package related to parental control watched content for a registered IMPU, the OITF with the appropriate parental control authority over the user shall issue a SIP SUBSCRIBE request to the IG that includes the SIP headers as per Table 27. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g. Via, Max-Forwards).

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP SUBSCRIBE request includes all mandatory SIP headers as per IETF RFC 3261 and the headers listed in Table 27. Following successful validation, the IG shall support the 3GPP IMS subscription to the event package as per ETSI TS 124 503. This includes inserting any additional mandatory SIP headers as required by ETSI TS 124 503.

The IG shall return to the OITF the response received from the network.

The IG shall ensure that the OITF is synchronized with the timer for refreshing the subscription as desired by the network.

When a SIP NOTIFY is received by the IG, the IG acting as a B2BUA in accordance with 7.3.2 shall validate that the body of the incoming SIP NOTIFY is compliant to XML schema associated with the event package before sending it to the OITF. The incoming NOTIFY shall include the SIP headers depicted in Table 29 and any other mandatory SIP headers as per IETF RFC 3261.

The OITF shall validate that the body of SIP NOTIFY is compliant to the XML schema associated with the parental control watched content event package. Following a successful validation, the OITF shall send a SIP 200 OK response to the IG. The SIP 200 OK response shall include the SIP headers depicted in Table 30 and any other mandatory SIP headers as per IETF RFC 3261.

The IG acting as a B2BUA in accordance with 7.3.2 shall first validate the SIP 200 OK response (or any other received response) before sending it to the network.

For all subsequent SIP NOTIFY requests received using the same SIP dialog, the IG acting as a B2BUA in accordance with 7.3.2 shall perform the same processing. The IG shall consider a subscription terminated if it is not renewed by the OITF.

7.3.3.4.1.2 Procedure for refreshing an existing subscription to parental control watched content event package (OITF-initiated)

The procedure for refreshing a subscription is the same as the procedure for initiating a subscription.

7.3.3.4.2 Parental control

7.3.3.4.2.1 Parental control blocking request – OITF-initiated

An OITF with proper parental control authority that desires to block a watched content for an IPTV end user under its authority shall initiate an instant message for that purpose by sending to the IG a SIP MESSAGE that includes all the SIP headers in Table 31 and any other mandatory SIP headers as per IETF RFC 3261. The content of the SIP MESSAGE shall include the body associated with “urn:oipf:application:iptv-parental-control”.

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 31. Following a successful validation, the IG shall generate a SIP MESSAGE request to the network conformant to IETF RFC 3428 and ETSI TS 124 503.

The IG acting as a B2BUA in accordance with subclause 7.3.2 shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 32 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.4.2.2 IPTV end-user OITF reception of a blocking request related to parental control

When the IG receives any SIP MESSAGE related to parental control, as indicated by the content type (being set to “urn:oipf:application:iptv-parental-control”) and destined for a user, the IG acting as a B2BUA in accordance with 7.3.2 shall validate that the incoming SIP MESSAGE is conformant to IETF RFC 3428 and ETSI TS 124 503 and includes all the mandatory headers as per IETF RFC 3261, ETSI TS 124 503 and all the headers listed in Table 33. Following the successful validation, the IG shall forward the SIP MESSAGE to the appropriate OITF.

The OITF shall act on the incoming SIP MESSAGE in accordance with 6.3.7.3.2, before it sends a response back to the IG.

When the SIP response is received from the OITF, the IG acting as a B2BUA in accordance with 7.3.2 shall forward it to the network. The response shall include the SIP headers as per IETF RFC 3261 and shall be validated by the IG before being sent to the network.

7.3.3.5 Network-based user notification services using SIP Notification request setup

To initiate a notification request, the OITF shall initiate a SIP MESSAGE to the IG that includes all the SIP headers in Table 34 and any other mandatory SIP headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The body of the SIP MESSAGE shall include the MIME type defined in 6.3.8.7.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 34. Following a successful validation, the IG shall generate a SIP MESSAGE request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 35 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.6 Content bookmarking using SIP

7.3.3.6.1 Storing content bookmarks

If the OITF decides to store a content bookmark for watched content during a multicast of unicast content streaming session, and if permitted by the network to do so, the OITF shall initiate a SIP INFO request to the IG based on the SIP INFO framework. The SIP INFO request shall include all the SIP headers in Table 38 and any other mandatory SIP headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The body of the SIP INFO message shall include the MIME type defined in 6.3.9.6.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INFO includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 38. Following a successful validation, the IG shall generate a SIP INFO request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 39 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.6.2 Network management of bookmark storage requests

At any time, the IG can receive a SIP UPDATE request from the network to remove or re-instate support for the reception of the CoD-Bookmark Info Package, according to 12 of 3GPP TS 26.237 from an OITF.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP UPDATE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 40. Following a successful validation, the IG shall forward the SIP UPDATE request to the OITF. The IG shall forward the received response to the network. A SIP 200 OK response shall include the SIP headers defined in Table 41 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.6.3 Content-related bookmark retrieval

The OITF shall follow the procedure defined in 7.3.3.2.2 for establishing a unicast streaming session. Following that, the IG shall receive a SIP INFO request from the network, based on the SIP INFO framework, and that includes the bookmarks list related to the requested content. The body of the SIP INFO request shall include the XML schema depicted in 6.3.9.6 and related to the CoD-Bookmark Info Package.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INFO request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 36. Following a successful validation, the IG shall forward the SIP INFO request to the OITF. The IG shall forward the received response to the network.

7.3.3.7 Local PVR service using SIP

7.3.3.7.1 OIPF-initiated PVR service capture request

To initiate a local PVR capture request, the OITF shall initiate a SIP MESSAGE to the IG that includes all the SIP headers in Table 42 and any other mandatory SIP headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The body of the SIP MESSAGE shall include the MIME type defined in 6.3.10.2.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 42. Following a successful validation, the IG shall generate a SIP MESSAGE request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 43 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.7.2 PVR service capture validation

At any time, the IG can receive a SIP MESSAGE request from the network related to an outstanding local PVR request.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 44. The body of the SIP MESSAGE request shall include the XML schema defined in 6.3.10.2. Following a successful validation, the IG shall forward the SIP INFO request to the OITF. The IG shall forward the received response to the network.

7.3.3.8 Network PVR (nPVR) using SIP

7.3.3.8.1 OIPF-initiated PVR service capture request

To initiate a network PVR capture request, the OITF shall initiate a SIP MESSAGE to the IG that includes all the SIP headers in Table 42 and any other mandatory SIP headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The body of the SIP MESSAGE shall include the MIME type defined in 6.3.10.2 with the qualifications listed in 6.3.11.1.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 42. Following a successful validation, the IG shall generate a SIP MESSAGE request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 43 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.8.2 PVR service capture outcome

At any time, the IG can receive a SIP MESSAGE request from the network reporting the outcome of a network PVR.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 44. The body of the SIP MESSAGE request shall include the XML schema defined in 6.3.11.2. Following a successful validation, the IG shall forward the SIP MESSAGE request to the OITF. The IG shall forward the received response to the network.

7.3.3.9 Session transfer using SIP

7.3.3.9.1 Generic procedures

7.3.3.9.1.1 Transferee session initiation procedures related to a transferred session

To initiate a session related to a transferred session, the transferee OITF shall initiate a SIP INVITE to the IG that includes all the headers as per Table 16 and other mandatory headers as per IETF RFC 3261 with the following additions:

- An additional SIP header – Replace header is included in the SIP INVITE request and is set to the appropriate information depending on the deployed mode. For the push mode, the information is retrieved from the incoming REFER request to the transferee as per Step 4 in 6.3.13.1.2.3.
- In the request Line, the wild card part (*) representing the content instance identifier to be transferred is constructed differently than specified in the table. In the push mode, this field is extracted from the To header field embedded in the Refer-To header in the incoming REFER request as per Step 4 in 6.3.13.1.2.3.

The OITF shall also include an SDP conformant to Step 1 as defined in 6.3.2.1.2. The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the request includes all the mandatory SIP headers as per IETF RFC 3261 and headers listed in Table 16. Furthermore, the IG perform the steps depicted in 7.3.3.9.1.2.

Following that, the IG acting as a B2BUA in accordance with 7.3.2 shall generate an initial SIP INVITE request to the network as specified in ETSI TS 124 503 for originating sessions.

The IG acting as a B2BUA in accordance with 7.3.2 shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 17 and all mandatory headers as per IETF RFC 3261. For a description of the received SDP answer, refer to 6.3.2.1.2.

When parsing the b=RR:<bandwidth-value> line in the received SDP answer: if the bandwidth value agreed is non-zero, then the transferee OITF shall send RTCP RRs and shall not send RTSP keep-alive messages. If the bandwidth value received is zero, then the transferee OITF shall not send RTCP RRs but instead it shall send RTSP keep-alive messages.

If the transferee OITF receives an error code with an Insufficient Bandwidth indication in the response from the IG, the OITF may perform a new SIP INVITE with a reduced maximum bandwidth for the content. This procedure may be repeated. If no agreement can be reached, the OITF may display a failure message to the user.

Finally to complete the SIP INVITE transaction, the transferee OITF shall send a SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 18 and all mandatory headers as per IETF RFC 3261.

7.3.3.9.1.2 IG handling of session transfers when the transferor and transferee are behind the same IG

Upon receipt by the IG acting as a B2BUA in accordance with 7.3.2 of a SIP INVITE related to a session transfer (as indicated by the presence of the Replace header), if the transferor and the transferee are behind the same IG, then during the session transfer procedure, the transferor OITF shall receive a SIP re-INVITE request from the IG and where the SIP headers are conformant to Table 16 with the qualification in Step 1 in 6.3.13.1.1.3. The SDP shall conform to Step 1 in 6.3.13.1.1.3 as well.

The OITF shall conform to Step 2 in 6.3.13.1.1.3 and subsequently shall send a SIP 200 OK response, and where the SIP headers are set according to Table 17 with the qualification in Step 2 in 6.3.13.1.1.3.

Subsequently, the transferor OITF shall receive a SIP ACK from the IG acting as a B2BUA in accordance with 7.3.2 conforming to Table 14 with the exception that the SIP headers are populated appropriately given that the ACK is sent from the IG to the OITF.

Following that, the IG shall perform Steps 4 and 5 as depicted in 6.3.13.1.1.3.

Note that this entire procedure is bypassed if the transferor and transferee are not behind the same IG.

7.3.3.9.2 Session transfer via push mode

7.3.3.9.2.1 Target discovery by the transferor OITF

An OITF that wants to locate a target OITF for session transfer purposes shall perform the procedures described in 7.3.4.2.4.

Subsequently a target OITF (transferee OITF) can be selected from the returned information.

7.3.3.9.2.2 Transferee OITF initiating a session transfer

A transferor OITF that desires to transfer a session to a transferee OITF shall initiate a SIP REFER request to the IG that includes all the SIP headers in Table 46 and any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The body of the SIP REFER message shall include the MIME type defined in 6.3.13.2.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP REFER includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 46. Following a successful validation, the IG shall generate a SIP REFER request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF. A SIP 202 OK response shall include the SIP headers defined in Table 47 and any other mandatory SIP headers as per IETF RFC 3261.

Later at some point in time, the IG shall receive a SIP NOTIFY request that reports the outcome of the session transfer request.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP NOTIFY request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 48. Following a successful validation, the IG shall forward the SIP NOTIFY request to the OITF. The IG shall forward the received response to the network. The SIP 200 OK response shall include the SIP headers in Table 49.

7.3.3.9.2.3 Transferee OITF handling of an incoming transfer

At some point in time, during a session transfer, the IG shall receive a SIP REFER request destined for a transferee OITF.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP REFER includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 46. The body of the SIP MESSAGE request shall include the XML schema defined in 6.3.13.2. Following a successful validation, the IG shall forward the SIP REFER request to the transferee OITF.

The transferee OITF shall examine the incoming SIP REFER request as per Step 2 in 6.3.13.1.2.3.

Once the transferee OITF accepts the incoming SIP REFER request, it shall send a SIP 202 OK response to the IG. The SIP 202 OK response shall include the SIP headers defined in Table 47 and all mandatory headers as per IETF RFC 3261. The IG shall validate the SIP 202 OK response before forwarding it to the network for delivery to the transferor OITF.

The transferee OITF shall then follow Step 4 in 6.3.13.1.2.3, and proceed to initiate a new session according to the procedure in 7.3.3.9.1.1.

Once the session setup outcome is determined, the transferee OITF shall initiate a SIP NOTIFY request to the IG that includes all the SIP headers in Table 48 and any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards).

The IG, acting as a B2BUA in accordance with 7.3.2, shall validate that the SIP NOTIFY includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 48. Following a successful validation, the IG shall generate a SIP NOTIFY request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the transferee OITF. A SIP 200 OK response shall include the SIP headers defined in Table 49 and all mandatory headers as per IETF RFC 3261.

7.3.3.10 Purchase of digital media service using SIP

7.3.3.10.1 Request initiation for purchase of digital media service

To initiate a request to purchase digital media selected by an end-user within a unicast content streaming session, the OITF shall initiate a SIP INFO request to the IG, based on SIP INFO framework, that includes all the SIP headers in Table 21 and any other mandatory SIP headers as per IETF RFC 3261 (e.g. Via, Max-Forwards). The SIP INFO request body shall include an XML document as defined in 6.3.5.9, which is associated with the Digital-Media-purchase Info Package. The initiation of the SIP INFO request assumes that the IPTV control FE indicated its willingness to receive the Digital-Media-purchase Info Package.

The IG, acting as a B2BUA shall validate that the SIP INFO request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 21. Following a successful validation, the IG shall generate a SIP INFO request as specified in ETSI TS 124 503 for originating UA.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 22 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.10.2 Network management of purchase requests of digital media

At any time, the IG can receive a SIP UPDATE request from the network to remove or reinstate support for the reception of the Digital-Media-purchase Info package from an OITF.

The IG, acting as a B2B UA shall validate that the SIP UPDATE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 23. Following a successful validation, the IG shall forward the SIP UPDATE request to the OITF. The IG shall forward the received response to the network. A SIP 200 OK response shall include the SIP headers defined in Table 24 and any other mandatory SIP headers as per IETF RFC 3261.

7.3.3.11 Personalized channel (network-centric model) using SIP Request initiation for personal channel

To initiate a session for a personalized channel, the OITF shall follow 7.3.3.2.2 with the difference that the content identifier in this case for both the request-URI and the To fields is set to the personalized channel identifier.

The IG, acting as a B2BUA in accordance with 7.3.2 shall also follow the same steps in 7.3.3.2.2.

7.3.4 Protocols for service access and control functions

7.3.4.1 Service provider discovery

To retrieve the list of service providers, the OITF shall issue a SIP SUBSCRIBE request to the IG that includes the SIP headers as per Table 50. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g., Via, Max-Forwards).

The OITF shall include a message body associated with the appid “urn:oipf:application:iptv-SP-discovery” representing the capabilities of the OITF.

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP SUBSCRIBE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 50. Following a successful validation, the IG shall follow the procedure in 7.2.4.1 when it comes to initiating a SIP SUBSCRIBE towards the service provider.

The IG acting as a B2BUA in accordance with 7.3.2 shall return to the OITF the response received from the network as per Step 3b in 7.2.4.1.

The IG shall ensure that the OITF is synchronized with the timer for refreshing the subscription as desired by the network.

The OITF is responsible for refreshing the subscription based on the received timer from the network. The IG shall consider a subscription terminated if it is not refreshed by the OITF before expiry.

The procedure for refreshing a subscription is the same as the procedure for initiating a subscription.

7.3.4.2 User registration and network authentication

7.3.4.2.1 Procedure for user registration

This procedure shall be invoked in the following cases:

- when the OITF is turned on;
- when an IPTV end user explicitly logs on at an OITF using an Alias or IMPU.

The IG shall extract the deviceId from the SIP instance feature tag.

If the deviceId and the IMPU match another deviceId and IMPU whose state is held in the IG, the IG shall conclude that the OITF has undergone a restart and shall proceed to immediately clear all SIP sessions belonging to the OITF. Following that, the IG shall de-register all users registered from that OITF.

If GRUU is not requested, the IG shall not perform IMS registration when the IMPU is already registered; however, the IG shall maintain a binding between the Alias/IMPU, the OITF device from which the registration is received (extracted from the sip instance feature tag), and the new contact information including the sip instance feature tag, which provides an easy way to guarantee uniqueness within the Address of Record (AOR).

If the identity being registered is not the default identity and if the default identity is not bound to any OITF in the consumer network, then the IG shall deregister the default identity at the end of this procedure.

If the identity being registered is the default identity, and if the default identity is not bound to any OITF in the consumer network, then the IG shall deregister its contact address for the default identity at the end of this procedure.

To register an identity, the OITF shall issue a SIP REGISTER request to the IG that includes the SIP headers as per Table 60. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g., Via, Max-Forwards).

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP REGISTER request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 60. Following a successful validation, the IG shall support the 3GPP IMS registration procedure as per ETSI TS 124 503. This includes handling of user authentication and authorization, and including any additional SIP headers that are mandatory based on ETSI TS 124 503.

Once the IG completes the IMS registration procedure, it shall return a SIP 200 OK to the OITF that includes the SIP headers listed in Table 61.

Following a successful registration, the IG shall maintain a binding between the Alias/IMPU, the OITF device from which the registration is received (extracted from the sip instance feature tag), and the new contact information including the sip instance feature tag for the duration of the registration.

7.3.4.2.2 Procedure for user deregistration

This procedure is invoked in the following cases:

- The OITF is turned off
- An IPTV end user, who has registered with his own IMPU, deregisters from an OITF

If GRUU is not supported for this registration, the IG shall not perform IMS deregistration when an IMPU is already registered on multiple OITFs, but the IG shall remove the binding between the IMPU and the OITF from which the user has deregistered (extracted from the sip instance feature tag) including the contact information (including the sip instance feature tag).

If GRUU is not supported for this registration, the IG shall perform the IMS deregistration procedure if the IMPU was bound to a single OITF.

Note that if following the successful de-registration of the IMPU, and if there are no more OITFs still turned on in the consumer network, the IG shall re-register the default identity from the IG point of view.

To de-register an identity, the OITF shall issue a SIP REGISTER request to the IG that includes the SIP headers as per Table 60. The OITF shall set the expires parameter to 0 for the contact to be de-registered. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g., Via, Max-Forwards).

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP REGISTER request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 60. Following a successful validation, the IG shall support the 3GPP IMS de-registration procedure as per ETSI TS 124 503. Once the IG completes the IMS de-registration procedure, it shall return a SIP 200 OK to the OITF that includes the SIP headers listed in Table 61.

Note that the OITF shall tear down and release all SIP sessions involving the contact to be de-registered prior to de-registering that contact. If the OITF does not tear down those SIP sessions before initiating the de-registration request to the IG, the IG shall tear down those sessions before performing the deregistration procedure described above.

Following a successful de-registration, the IG shall remove the binding between the Alias/IMPU, the OITF device from which the registration is received (extracted from the sip instance feature tag).

7.3.4.2.3 Procedure for refreshing a registration

This procedure shall be initiated by the OITF at any time before the expiry of the registration refresh timer.

The procedure is the same as the procedure for registering a user. A registration shall be terminated if it is not refreshed before the expiry of the registration refresh timer.

For an OITF-initiated registration, the IG shall consider a registration terminated (that is the user de-registered) if it is not refreshed. In this case, the IG executes the procedures associated with user deregistration.

7.3.4.2.4 Procedure for subscription to the registration-state event package

This procedure shall be invoked by the OITF immediately after the successful registration of an IMPU.

To subscribe to the registration-state event package for the successfully registered IMPU, the OITF shall issue a SIP SUBSCRIBE request to the IG that includes the SIP headers as per Table 62. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g., Via, Max-Forwards).

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP SUBSCRIBE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 62. Following a successful validation, the IG shall support the 3GPP IMS subscription to

registration-event package as per ETSI TS 124 503. This includes inserting any additional SIP headers that are mandatory based on ETSI TS 124 503.

The IG acting as a B2BUA in accordance with 7.3.2 shall return to the OITF the response received from the network.

The IG shall ensure that the OITF is synchronized with the timer for refreshing the subscription as desired by the network.

When a SIP NOTIFY request is received by the IG, the IG acting as a B2BUA in accordance with 7.3.2 shall validate the incoming NOTIFY request before sending it to the OITF. The incoming NOTIFY request shall include the SIP headers depicted in Table 64 and any other mandatory headers as per IETF RFC 3261.

The OITF shall validate that the SIP body is compliant to the XML schema associated with the registration-state event package. Following a successful validation, the OITF shall send a SIP 200 OK response to the IG. The SIP 200 OK response shall include the SIP headers depicted in Table 65. The IG acting as a B2BUA in accordance with 7.3.2 shall first validate the SIP 200 OK response (or any other received response) before sending it to the network.

For all subsequent SIP NOTIFY requests using the same SIP dialog, the IG acting as a B2BUA in accordance with 7.3.2 shall perform the same processing.

7.3.4.2.5 Procedure for terminating a subscription to the registration-state event package

The OITF is responsible for refreshing the subscription based on the received timer from the network. The IG shall consider a subscription terminated if it is not refreshed by the OITF before expiry.

The procedure for refreshing a subscription is the same as the procedure for initiating a subscription.

7.3.4.3 Notification of service profile changes

7.3.4.3.1 Subscription to notifications of service profile changes (xcap-diff)

To subscribe for the purpose of receiving notification of changes in a service profile, the OITF shall issue a SIP SUBSCRIBE request to the IG that includes the SIP headers as per Table 54. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g., Via, Max-Forwards). The body of the SIP SUBSCRIBE request shall include the list of the requested URIs associated with the XCAP resources for which the subscription is issued. The MIME Type of the document inserted in the body will be signalled by the Content-Type header set to "application/vnd.oipf.userprofile+xml".

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP SUBSCRIBE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 54. Following successful validation, the IG shall support the 3GPP IMS subscription to event packages and shall send a SUBSCRIBE request to the IPTV service profile FE in accordance with ETSI TS 124 503, and IETF RFC 5875.

The IG acting as a B2BUA in accordance with 7.3.2 shall return to the OITF the response received from the network.

The IG shall ensure that the OITF is synchronized with the timer for refreshing the subscription as desired by the network.

When a SIP NOTIFY request is received by the IG, the IG acting as a B2BUA in accordance with 7.3.2 shall validate the incoming NOTIFY request includes the SIP headers depicted in Table 56 and any other mandatory headers as per IETF RFC 3261.

The OITF shall validate that the SIP body is compliant to the XML schema associated with application/XCAP-diff+xml as defined in IETF RFC 5875 and IETF RFC 5874. Following a successful validation, the OITF shall send a SIP 200 OK response to the IG. The SIP 200 OK response shall include the SIP headers depicted in Table 57. The IG acting as a B2BUA in accordance with 7.3.2 shall first validate the SIP 200 OK response (or any other received response) before sending it to the network.

For all subsequent SIP NOTIFY requests received using the same SIP dialog, the IG acting as a B2BUA in accordance with 7.3.2 shall perform the same processing. The IG shall consider a subscription terminated if it is not renewed by the OITF.

7.3.4.3.2 Procedure for refreshing a subscription to receive notifications of service profile changes over the HNI-IGI SIP option

The procedure for refreshing a subscription is the same as the procedure for initiating a new subscription.

7.3.5 Protocols for communication services using SIP

7.3.5.1 Procedure for instant message based CallerID

The procedure for caller id is identical to the reception of an instant message as depicted in subclause 7.3.5.2.2.

7.3.5.2 Instant messaging

7.3.5.2.1 Procedure for outgoing instant messaging

To initiate an instant message, the OITF shall send a SIP MESSAGE to the IG that includes all the SIP headers in Table 71 and any other mandatory headers as per IETF RFC 3261. The content of the SIP MESSAGE shall be conformant to IETF RFC 3428.

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP MESSAGE includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 71. Following a successful validation, the IG shall generate a SIP MESSAGE request to the network conformant to OMA-ERP-Presence_SIMPLE-V1_1-20080627-A.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 72 and all mandatory headers as per IETF RFC 3261.

7.3.5.2.2 Procedure for incoming instant messaging

When the IG receives any SIP MESSAGE destined for a user, the IG acting as a B2BUA in accordance with 7.3.2 shall validate that the incoming SIP MESSAGE is conformant to OMA-ERP-Presence_SIMPLE-V1_1-20080627-A and includes all the mandatory headers as per IETF RFC 3261 and all the headers listed in Table 73. Following the successful validation, the IG shall forward the SIP MESSAGE to the appropriate OITF.

When the SIP response is received from the OITF, the IG acting as a B2BUA in accordance with 7.3.2 shall forward it to the network. The response shall include the SIP headers in Table 72 and shall be validated by the IG before being sent to the network.

7.3.5.3 Presence

7.3.5.3.1 Procedure for subscribing to presence information

To subscribe to the presence event package for any registered IMPU, the OITF shall issue a SIP SUBSCRIBE request to the IG that includes the SIP headers as per Table 90. In addition, the OITF shall include all mandatory SIP headers as per IETF RFC 3261 and that are missing from the table (e.g. Via, Max-Forwards).

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP SUBSCRIBE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 90. Following successful validation, the IG shall support the 3GPP IMS subscription to presence event package as per OMA-ERP-Presence_SIMPLE-V1_1-20080627-A. This includes inserting any additional SIP headers that are mandatory.

The IG shall return to the OITF the response received from the network.

The IG shall ensure that the OITF is synchronized with the timer for refreshing the subscription as desired by the network.

When a SIP NOTIFY request is received by the IG, the IG shall validate the incoming NOTIFY request being compliant to OMA-ERP-Presence_SIMPLE-V1_1-20080627-A before sending it to the OITF. The incoming NOTIFY request shall include the SIP headers depicted in Table 92 and any other mandatory headers as per IETF RFC 3261.

The OITF shall validate that the SIP body is compliant to the XML schema associated with the presence event package. Following a successful validation, the OITF shall send a SIP 200 OK response to the IG. The SIP 200 OK response shall include the SIP headers depicted in Table 93. The IG shall first validate the SIP 200 OK response (or any other received response) before sending it to the network.

For all subsequent SIP NOTIFY requests received using the same SIP dialog, the IG acting as a B2BUA in accordance with 7.3.2 shall perform the same processing. The IG shall consider a subscription terminated if it is not renewed by the OITF.

7.3.5.3.2 Procedure for terminating a subscription to presence information

The procedure for refreshing a subscription is the same as the procedure for initiating a subscription.

7.3.5.3.3 Procedure for publishing presence information

To publish presence information, the OITF shall send a SIP PUBLISH request to the IG that includes all the SIP headers in Table 94 and other mandatory headers as per IETF RFC 3261. The content of the SIP PUBLISH shall be based on subclause 6.5.4.6.

The IG acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP PUBLISH request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 94. Following a successful validation, the IG shall generate a SIP PUBLISH request to the network conformant to OMA-ERP-Presence_SIMPLE-V1_1-20080627-A.

The IG shall forward any received SIP response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 95 and all mandatory headers as per IETF RFC 3261.

7.3.5.4 Chatting

7.3.5.4.1 General

For IM sessions, the OITF shall implement the MSRP stack conformant to OMA-TS-SIMPLE_IM-V1_0-20100322-C.

7.3.5.4.2 IM session initiation

To initiate an IM session, the OITF shall initiate a SIP INVITE request to the IG that includes all the SIP headers in Table 74. However, the body of the SIP INVITE shall be populated with the following information to initiate the MSRP session:

- a c = IN IP4 <IP address> , where <IP address> would contain the IP address of the OITF;
- an m = message <tcp port> tcp/msrp, where tcp port is a TCP port could be set to the dummy value "9";
- an a = accept-types:message/cpim, attribute which is mapped from the "X-OITF-Accept:" header value;
- an a = path msrp://<IP address>:<tcpport>/<session-id>; tcp, where:
 - <IP address> would contain the IP address of the OITF;
 - <tcpport> would be assigned automatically by the OITF;
 - <session-id> would be assigned by the OITF and bound to the requesting OITF IM SIP chatting application.

The OITF shall also include any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards).

The IG, acting as a B2BUA in accordance with subclause 7.3.2 shall validate that the SIP INVITE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 74. Following a successful validation, the IG shall generate an initial INVITE request as specified in OMA-TS-SIMPLE_IM-V1_0-20100322-C.

The IG acting as a B2BUA in accordance with 7.3.2 shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 75 and all mandatory headers as per IETF RFC 3261.

The OITF shall save the information returned in the SDP for the purpose of MSRP.

Finally to complete the INVITE transaction, the OITF shall then send a SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 76 and all mandatory headers as per IETF RFC 3261.

7.3.5.4.3 Incoming IM session

When the IG receives a SIP INVITE request related to an IM session and destined for a user, the IG acting as a B2BUA in accordance with subclause 7.3.2 shall validate that the incoming SIP INVITE is conformant to OMA-TS-SIMPLE_IM-V1_0-20100322-C and includes all the mandatory headers as per IETF RFC 3261 and all the headers listed in Table 87. Following the successful validation, the IG shall forward the SIP INVITE to the appropriate OITF including the received SDP.

The OITF shall validate that the SDP includes all the necessary and mandatory parameters per IETF RFC 4975. The OITF shall store the necessary information and shall send its response to the IG. The SDP included in the response shall include the following information:

- a c = IN IP4 <IP address> , where <IP address> would contain the IP address of the OITF;

- an m = message <tcp port> tcp/msrp, where tcp port is a TCP port could be set to the dummy value "9";
- an a = accept-types:message/cpim, attribute;
- an a = path msrp://<IP address>:<tcpport>/<session-id>; tcp, where:
 - <IP address> would contain the IP address of the OITF;
 - <tcpport> would be assigned automatically by the OITF;
 - <session-id> would be assigned by the OITF and bound to the responding OITF SIP chatting application.

When the SIP response is received from the OITF, the IG acting as a B2BUA in accordance with subclause 7.3.2 shall forward it to the network. The response shall include the SIP headers in Table 88 and shall be validated by the IG before being sent to the network.

Finally to complete the INVITE transaction, the IG shall forward the received SIP ACK from the network to the OITF. The SIP ACK shall include the SIP headers defined in Table 89 and all mandatory headers as per IETF RFC 3261.

7.3.5.4.4 Session termination

To terminate a session, the OITF shall send a SIP BYE request to the IG that includes all headers listed in Table 83 and any other mandatory headers as per IETF RFC 3261.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP BYE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 83. Following a successful validation, the IG shall generate a SIP BYE request as specified in OMA-TS-SIMPLE_IM-V1_0-20100322-C. The IG shall forward the received response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 84 and all mandatory headers as per IETF RFC 3261.

Alternatively, on receipt of a SIP BYE request from the IPTV control FE, the IG acting as a B2BUA in accordance with 7.3.2, after validating the request, to include all headers listed in Table 85 and any other mandatory headers as per IETF RFC 3261, shall forward the request to the OITF. The OITF shall respond with a SIP 200 OK response, which the IG forward to the IPTV control FE.

7.3.5.4.5 Session refresh

It is the responsibility of the OITF to refresh the IM session, as per IETF RFC 3556 before the session expires. The IG shall consider a session terminated if it is not refreshed, and shall clear its internal state accordingly.

7.3.5.4.6 Outgoing/incoming MSRP chat and chat state messages

Following the successful establishment of an incoming or outgoing IM session, the OITF is ready to receive or send any MSRP chat message or MSRP chat state message based on IETF RFC 4975 and OMA-TS-SIMPLE_IM-V1_0-20100322-C.

7.3.5.5 Content sharing

7.3.5.5.1 Procedure for outgoing content sharing

The OITF can initiate a content sharing session and can initiate a change request to transfer the session to other OITF or to maintain an existing session between two sides.

To initiate a content sharing session, the OITF shall send a SIP INVITE to the IG that includes all the SIP headers and any other mandatory headers as per IETF RFC 3261. SDP shall be used as specified in ETSI TS 124 503.

The IG shall validate that the SIP INVITE includes all the mandatory headers as per IETF RFC 3261. Following a successful validation, the IG shall generate a SIP INVITE request to the network conformant to IETF RFC 3261.

An REFER message associated with transferor OITF shall be sent to the IG that includes all the SIP headers and any other mandatory headers as per IETF RFC 3261.

- The IG shall forward any incoming SIP REFER, for the purpose of session transfer, that conforms to 3GPP TS 24.237 to the transferee OITF. Non-conformant SIP REFER messages shall be rejected with the appropriate response code.
- An outgoing SIP NOTIFY message associated with the transferee reporting the outcome of a session transfer associated with a REFER request shall conform to IETF RFC 3515.
- The IG shall forward any incoming SIP NOTIFY, reporting the outcome of a session transfer associated with a REFER request, that conforms to IETF RFC 3515 to the transferor OITF. Non-conformant SIP REFER messages shall be rejected with the appropriate response code.

When receiving a SIP REFER request, the IPTV control FE shall authorize the request. If the user is not authorized to perform session transfer, an appropriate SIP error response is returned. If the user is authorized, the IPTV control FE shall forward the SIP REFER to the ASM for delivery to its destination (transferee OITF). The IPTV control FE shall be stateful to the session transfer procedure

When receiving a SIP NOTIFY associated with a SIP REFER request, the IPTV control FE shall validate the SIP NOTIFY. If not successfully validated, an appropriate SIP response shall be returned. If successfully validated, the IPTV control FE shall forward the SIP NOTIFY to the ASM for delivery to the destination (transferor OITF).

7.3.5.5.2 Procedure for incoming content sharing

When the IG receives a SIP INVITE request related to a content sharing session and destined for a user, the IG shall validate that the incoming SIP INVITE is conformant to OMA-TS-SIMPLE_IM-V1_0-20100322-C and includes all the mandatory headers as per IETF RFC 3261. Following the successful validation, the IG shall forward the SIP INVITE to the appropriate OITF including the received SDP

- The OITF shall validate that the SDP includes all the necessary and mandatory parameters per ETSI TS 124 503. The OITF shall store the necessary information and shall send its response to the IG.

When the SIP response is received from the OITF, the IG shall forward it to the network.

7.3.5.5.3 Procedure for terminating content sharing

To terminate a session, the OITF shall send a SIP BYE request to the IG that includes all the mandatory headers as per IETF RFC 3261.

The IG, acting as a B2B UA shall validate that the SIP BYE request includes all the mandatory headers as per IETF RFC 3261. Following a successful validation, the IG shall generate a SIP BYE request as specified in IETF RFC 3261. The IG shall forward the received response to the OITF. A SIP 200 OK response shall include all the mandatory headers as per IETF RFC 3261.

Alternatively, on receipt of a SIP BYE request from the IPTV control FE, the IG, after validating the request, to include all the mandatory headers as per IETF RFC 3261, shall forward the request to the OITF. The OITF shall respond with a SIP 200 OK response, which the IG forward to the IPTV control FE.

7.3.5.6 Multimedia (MM) telephony

7.3.5.6.1 OITF-initiated multimedia telephony session initiation

To initiate a MM telephony session, the OITF shall initiate a SIP INVITE request to the IG that includes all the SIP headers in Table 134. The body of the SIP INVITE shall be populated in accordance with IETF RFC 4566 reflecting the appropriate description for each media, and corresponding codecs for MM session. Subclause 7.3.5.6.8 details the SDP representation for codecs that can be used by the OITF.

The OITF shall also include any other mandatory headers as per IETF RFC 3261 (e.g. Via, Max-Forwards).

The IG, being service aware, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP INVITE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 134. From the included codecs and requested media, the IG shall be able to conclude that the requested session is for MMTEL. Following a successful validation, and detection of the implied MMTEL service for the incoming session, the IG shall generate an initial INVITE request towards the ASM as specified in ETSI TS 124 503 and ETSI TS 181 005 and shall include all mandatory headers.

The IG acting as a B2BUA in accordance with 7.3.2 shall forward any received SIP response to the OITF including the information in the SDP. A SIP 200 OK response shall include the SIP headers defined in Table 135 and all mandatory headers as per IETF RFC 3261.

Finally to complete the INVITE transaction, the OITF shall then send a SIP ACK to the IG. The SIP ACK shall include the SIP headers defined in Table 136 and all mandatory headers as per IETF RFC 3261. The IG acting as a B2BUA, in accordance with 7.3.2, shall forward the SIP ACK to the ASM.

7.3.5.6.2 Incoming multimedia telephony session initiation

When the IG receives a SIP INVITE request related to an MMTEL session and destined for a user, the IG shall validate that the incoming SIP INVITE is conformant to ETSI TS 124 503 and ETSI TS 181 005 and includes all the mandatory headers as per IETF RFC 3261 and all the headers listed in Table 137. Following the successful validation, the IG acting as a B2BUA in accordance with 7.3.2 shall forward the SIP INVITE to the appropriate OITF including the received SDP

The OITF shall validate that the SDP includes all the necessary and mandatory parameters per ETSI TS 124 503 and ETSI TS 181 005 and the mandatory information pertinent to the chosen codecs in accordance with 7.3.5.6.8. The OITF shall store the necessary information and shall send its response to the IG. The SDP included in the response shall include what is acceptable to the OITF based on the received SDP per ETSI TS 124 503 and ETSI TS 181 005.

When the SIP response is received from the OITF, the IG acting as a B2BUA in accordance with 7.3.2, shall forward it to the network. The response shall include the SIP headers in Table 138 and shall be validated by the IG before being sent to the network.

Finally to complete the INVITE transaction, the IG acting as a B2BUA in accordance with 7.3.2 shall forward the received SIP ACK from the network to the OITF. The SIP ACK shall include the SIP headers defined in Table 139 and all mandatory headers as per IETF RFC 3261.

7.3.5.6.3 OITF-initiated session termination

To terminate a session, the OITF shall send a SIP BYE request to the IG that includes all headers listed in Table 140 and any other mandatory headers as per IETF RFC 3261.

The IG, acting as a B2BUA in accordance with 7.3.2 shall validate that the SIP BYE request includes all the mandatory headers as per IETF RFC 3261 and the headers listed in Table 140. Following a successful validation, the IG shall generate a SIP BYE request as specified in ETSI TS 124 503 and ETSI TS 181 005 towards the ASM. The IG acting as a B2BUA in accordance with 7.3.2 shall forward the received response to the OITF. A SIP 200 OK response shall include the SIP headers defined in Table 141 and all mandatory headers as per IETF RFC 3261.

7.3.5.6.4 Incoming session termination

Upon receipt of a SIP BYE request from the network, the IG, after validating the request to include all headers listed in Table 142 and any other mandatory headers as per IETF RFC 3261, and acting as a B2BUA in accordance with 7.3.2 shall forward the request to the OITF.

The OITF shall respond with a SIP 200 OK response. The SIP 200 OK response shall include the headers in Table 143, and which the IG, acting as a B2BUA in accordance with 7.3.2 shall forward to the network.

7.3.5.6.5 Session modification

OITF initiated session modification or an incoming session modification request is essentially identical to session initiation and an incoming session initiation respectively from a handling prospective, both for the OITF and the IG.

7.3.5.6.6 Session refresh

It is the responsibility of the OITF to refresh the MM session, as per IETF RFC 3556 before the session expires. The IG shall consider a session terminated if it is not refreshed, and shall clear its internal state accordingly.

7.3.5.6.7 Tables of message elements

Table 134 – Supported SIP headers in the HNI-IGI INVITE request message for an outgoing MM telephony session setup (OITF→IG)

SIP header	Source of information for coding purposes
request-Line The request-URI in the INVITE request shall be set to the called identity.	IETF RFC 3261 INVITE <request URI> SIP/2.0
From	IETF RFC 3261
To The URI part of To shall be set to the value of the request URI in the “request-Line”	IETF RFC 3261
Contact If GRUU has not been requested at registration, then the URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request. If GRUU has been requested at registration, then the OITF shall include in the contact header the returned GRUU during the registration process. The IG includes all other mandatory parameters that are absent. Expires parameter should be included	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Content-Type	IETF RFC 3261 (application/sdp)
Content-Length	IETF RFC 3261
Supported If GRUU is used, Supported shall also be set to “gruu”	IETF RFC 3261 set to timer
session-Expires	IETF RFC 4028, IETF RFC 3556

Table 135 – Supported SIP headers in the response message to an HNI-IGI INVITE request message for an outgoing MM telephony session setup (IG→OITF)

SIP header	Source of information for coding purposes
response-Line	IETF RFC 3261 SIP/2.0 <response>
From	IETF RFC 3261
To	IETF RFC 3261
Contact	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
session-Expires	IETF RFC 4028, IETF RFC 3556
Content-Type	IETF RFC 3261
Content-Length	IETF RFC 3261

Table 136 – Supported SIP headers in the HNI-IGI ACK message for a successful MM telephony session setup (OITF→IG)

SIP header	Source of information for coding purposes
request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message	IETF RFC 3261 ACK <request URI> SIP/2.0
From	IETF RFC 3261
To The URI part of To shall be set to the value of the request URI in the "request-Line" of the initial request	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Contact The URI parameter shall be included, and shall match what has been inserted in the INVITE message. The IG includes all other mandatory parameters that are absent.	IETF RFC 3261

Table 137 – Supported SIP headers in the HNI-IGI INVITE request message for an incoming MM telephony session setup (IG→OITF)

SIP header	Source of information for coding purposes
request-Line The request-URI in the INVITE request shall be set to the called identity (the user logged in user IMPU on an OITF).	IETF RFC 3261 INVITE <request URI> SIP/2.0
From	IETF RFC 3261
To The URI part of To shall be set to the value of the request URI in the "request-Line".	IETF RFC 3261
Contact	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Content-Type	IETF RFC 3261 (application/sdp)
Content-Length	IETF RFC 3261
Supported	IETF RFC 3261 set to timer, and/or GRUU if supported by remote user
session-Expires	IETF RFC 4028, IETF RFC 3556

Table 138 – Supported SIP headers in the response message to an HNI-IGI INVITE request message for an incoming MM telephony session setup (OITF→IG)

SIP header	Source of information for coding purposes
response-Line	IETF RFC 3261 SIP/2.0 <response>
From	IETF RFC 3261
To	IETF RFC 3261
Contact If GRUU has not been requested at registration, then the URI parameter and the sip.instance feature tag shall be included and shall match what is sent in the contact header included in the registration request. If GRUU has been requested at registration, then the OITF shall include in the contact header the returned GRUU during the registration process. Expires parameter should be included The IG includes all other mandatory parameters that are absent.	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
session-Expires	IETF RFC 4028, IETF RFC 3556
Content-Type	IETF RFC 3261
Content-Length	IETF RFC 3261

Table 139 – Supported SIP headers in the HNI-IGI ACK message for a successful incoming MM telephony session setup (IG→OITF)

SIP header	Source of information for coding purposes
request-Line The request-URI in the ACK request shall be the contact included in the response to the INVITE message.	IETF RFC 3261 ACK <request URI> SIP/2.0
From	IETF RFC 3261
To The URI part of To shall be set to the value of the request URI in the "request-Line" of the initial request.	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Contact The URI parameter shall be included, and shall match what has been included in the initial INVITE message.	IETF RFC 3261

Table 140 – List of SIP headers for an outgoing SIP BYE (OITF→IG)

SIP header	Source of coding information
request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP INVITE (for outgoing session) or a 200 OK (for incoming session)	IETF RFC 3261 BYE <request URI> SIP/ 2.0
From	IETF RFC 3261
To	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Content-Length Shall be set to 0	IETF RFC 3261
Contact	IETF RFC 3261

Table 141 – List of SIP headers for the response to an outgoing SIP BYE (IG→OITF)

SIP header	Source of coding information
response-Line	IETF RFC 3261 SIP/2.0 <response>
From	IETF RFC 3261
To	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261

Table 142 – List of SIP headers for an incoming SIP BYE (IG→OITF)

SIP header	Source of coding information
request-Line Note that the request URI shall match the contact URI included in the contact field of the SIP INVITE (for outgoing session) or a 200 OK (for incoming session)	IETF RFC 3261 BYE <request URI> SIP/ 2.0
From	IETF RFC 3261
To	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261
Content-Length Shall be set to 0	IETF RFC 3261
Contact	IETF RFC 3261

Table 143 – List of SIP headers for the response to an incoming SIP BYE (OITF→IG)

SIP header	Source of coding information
response-Line	IETF RFC 3261 SIP/2.0 <response>
From	IETF RFC 3261
To	IETF RFC 3261
Call-ID	IETF RFC 3261
CSeq	IETF RFC 3261

7.3.5.6.8 SDP media type parameters for media description

This subclause describes the media configuration in SDP for all audio and video codecs defined in IEC 62766-2-1 for multimedia telephony services.

- Media type parameters to configure H.264 video RTP payload types shall comply with IETF RFC 3984.
- Media type parameters to configure MPEG-4 Part-2 video RTP payload types shall comply with IETF RFC 3016.
- Media type parameters to configure H.263 video RTP payload types shall comply with IETF RFC 4627.
- Media type parameters to configure G.711 or G.722 audio RTP payload types shall comply with IETF RFC 3551.
- Media type parameters to configure AMR-NB or AMR-WB RTP payload types shall comply with IETF RFC 4867.
- Media type parameters to configure G.719 payload types shall comply with IETF RFC 5404.
- Media type parameters to configure MPEG-4 AAC-LD/-ELD payload types shall comply with IETF RFC 3640.

NOTE IETF RFC 3640 defines transport and signalling for all MPEG-4 media types. To use it for MPEG-4 AAC-LD/-ELD audio, the “streamType” is set to “5” to signal an audio stream and the “mode” is either set to “aac-hbr” or “aac-lbr”. The hexadecimal value of the “config” parameter is the AudioSpecificConfig(), as defined in ISO/IEC 14496-3. The AudioSpecificConfig() contains the audioObjectType, that shall be set to “39” for AAC-ELD and “23” for AAC-LD.

EXAMPLE 1 SDP for AAC-LD, stereo 48 kHz:

```
m=audio 49230 RTP/AVP 96
```

```
a=rtpmap:96 mpeg4-generic/48000/2
```

```
a=fmtp:96 streamtype=5; profile-level-id=52; mode=AAC-hbr; config=11B0; sizeLength=13; indexLength=3; indexDeltaLength=3; constantDuration=1024
```

NOTE The a=fmtp line comprises a single line in the SDP file.

EXAMPLE 2 SDP for G.719, stereo 48kHz:

```
m=audio 49230 RTP/AVP 96
```

```
a=rtpmap:96 G719/48000/2
```

8 RTSP

8.1 General

Clause 8 defines the protocol for the use of RTSP over the following reference points:

- UNIS-11
- NPI-10

8.2 Protocols for IPTV service functions

8.2.1 Use of RTSP for unicast content streaming services

8.2.1.1 RTSP profile without SIP session management over UNIS-11 and NPI-10

8.2.1.1.1 General

The RTSP protocol shall be used on UNIS-11 and NPI-10 for unicast service setup and delivery. The OITF shall obtain an RTSP request URL from the content guide, prior to delivery of the media from a cluster controller. The use of RTSP shall comply with IETF RFC 2326 and with the following profile.

The following describes the RTSP profile for this Release. The functionalities not identified in this subclause are out of scope of OIPF:

- NPT Range time format shall be supported by clients and servers, Subclause 3.6 of IETF RFC 2326.
- The extension mechanism using option tags in Subclause 3.8 of IETF RFC 2326 shall not be used.
- Since in this Release is constraint to one media stream per session (one m= line of audio/video data), then:
 - this profile is not multi-server capable (Subclause 1.4 of IETF RFC 2326);
 - this profile does not support aggregate control. Aggregated support allows to control several associated streams (e.g., video and sound track) using one RTSP URI;
 - this profile does not support “pipelining” of RTSP messages, see 9.1 of IETF RFC 2326, “Pipelining”;
 - this profile shall support the following MIME types:
 - SDP (application/sdp) as the normative session description format, and
 - the MIME type text/parameters for GET_PARAMETER.
- Servers shall not use RTSP proxy features.
- Servers shall not use encryption or authentication.

The RTSP client shall understand the class of each status code, i.e., 1xx Information, 2xx Success, 3xx Redirection, 4xx Client Error, and 5xx Server Error. Additionally, a client shall understand the following status codes:

- "200": OK
- "301": Moved Permanently
- "302": Moved Temporarily
- "400": Bad request
- "404": Not Found
- "405": Method Not Allowed
- "408": request Time-out
- "462": Destination Unreachable
- "500": Internal Server Error
- "501": Not Implemented

Servers shall not use interleaved RTP and RTSP over TCP, as per 10.12, “Embedded (Interleaved) Binary Data” of IETF RFC 2326.

Clients shall not use PLAY messages in the PLAY state as keep-alives (10.5 of IETF RFC 2326).

Servers shall not use RTSP over UDP, see Appendix G of IETF RFC 2326 and related functionality like the rtspu URI scheme in 22.14.3 of IETF RFC 2326.

Regarding the RTSP header definitions, the client shall support the following headers:

- Accept, Allow, Content-Type, CSeq, Location, Public, Range, RTP-Info, Scale, session, Transport

Additionally, the following clarifications and best practices are added:

- RTSP URIs shall comply to the encodings and escape conventions defined in IETF RFC 3986, which extends the encodings and conventions that were hinted at but not specified in IETF RFC 2326.
- Regarding Clause 9 of IETF RFC 2326, it is recommended that servers use persistent TCP connections. This reduces session management complexity (keep-alive, tear down, etc) in the clients and the servers. Clients should use persistent connections.

Regarding keep-alive mechanisms, the following mechanisms are recommended in this order:

- if the OITF has agreed to send RTCP packets, it is recommended that these be re-used to keep the RTSP session alive;
- if not, it is recommended that the same empty RTP packets with payload type value 20 used for NAT traversal (see E.5.4) be re-used to keep the RTSP session alive;
- otherwise, the SET_PARAMETER Method (or GET_PARAMETER, whichever is supported) with an empty body should be used.

Finally, the RTSP OPTIONS Method should be used.

Regarding subclause 12.17, "CSeq", of IETF RFC 2326, the current best practice rules and clarifications are added were not clear in RTSP, in particular:

- if a server returns a "400 Bad request" response because the client did not include a CSeq header, then the response shall not include a CSeq header;
- the CSeq header value shall be increased by one for each new RTSP request;
- It is recommended that CSeq value starts at "1" (one).

Regarding non-seekable streams, Annex C.1.5 of IETF RFC 2326, these are indicated by using open-ended time intervals via a=range attribute in SDP. For example, "a=range:npt=0-".

Regarding Content-Length, an RTSP message with a message body shall include the Content-Length header. This can be misinterpreted from 4.3 and 12.14 of IETF RFC 2326, because 4.3 of IETF RFC 2326 refers to HTTP/1.1 (which only recommends it) while 12.14 of IETF RFC 2326 clearly says that it shall be included.

Regarding RTP-Info, 12.33 of IETF RFC 2326 states:

- The client shall not use the RTP SSRC parameter ("ssrc:") in a SETUP request. This is deprecated because it incompatible with the specification of RTP in IETF RFC 3550.
- The URL values in the "url=" parameter shall be quoted, e.g.: RTP-Info: url="rtsp://live.example.com/concert/audio". This allows the URL to contain ";" and "," characters.
- If the value of the "url" parameter in RTP-Info header is a relative URI, then the request-URI is used as base-URI. If it is an absolute URI, then this URI is the same as in the SETUP message.

8.2.1.1.2 RTSP session setup

When performing RTSP session setup, the OITF shall use the request URL to send a DESCRIBE message to the cluster controller to obtain a media SDP. The DESCRIBE message shall include the Accept header with the application/sdp content type. The cluster controller or CDF shall return a Content-Type header with application/sdp and the format of the body shall be according to IETF RFC 4566, unless there is redirection of DESCRIBE or SETUP. The OITF shall then issue a SETUP request to the cluster controller.

If the cluster controller can handle the request, the DESCRIBE and SETUP messages shall be forwarded to the most appropriate CDF.

If the cluster controller cannot handle the request, the cluster controller shall reply with a redirect response (Moved) message containing a URL of another cluster controller. The redirection may occur when receiving either a DESCRIBE or a SETUP.

If the cluster controller sends a redirect response to the SETUP request, the OITF may send a new DESCRIBE request.

If the setup is successful the CDF shall reply with a 200 OK message that shall be proxied by the cluster controller to the OITF. After receiving the setup response the OITF may send PLAY and PAUSE messages to the cluster controller.

The cluster controller shall modify the RTSP URL to forward the RTSP messages to the chosen CDF function, when the messages are initiated from the OITF.

When receiving error messages from the CDF, the cluster controller shall either forward them to the OITF or try another CDF.

When the OITF receives an error message, it may display appropriate messages to the end user. The error messages may also be handled by the downloaded DAE or native application before being displayed to the user.

8.2.1.1.3 RTSP Control for media delivery

8.2.1.1.3.1 Handling of media control for starting playback

On receiving a request from the user to start playback, the OITF shall send an RTSP PLAY message to the cluster controller.

The RTSP fields in the RTSP PLAY message shall be filled as follows:

On receiving a request from the user to modify the playback, the OITF shall send an RTSP PLAY message with a request to modify the position, speed and/or direction of playback. The OITF indicates the direction and/or speed of playback by including a Scale header or changes the position of playback by including a Range header.

The Scale header shall be set as follows:

- 1 indicates normal play;
- if not 1, the value corresponds to the rate with respect to normal viewing rate;
- a negative value indicates reverse direction.

If the request is to pause the playback, the OITF shall send an RTSP PAUSE message.

On receipt of a RTSP PLAY or PAUSE request, the cluster controller shall forward the message to the chosen CDF.

The CDF shall respond with a 200 OK message. The contents of the 200 OK response shall be as follows:

- CSeq shall be set to the same value as that in the request.

8.2.1.1.3.2 Handling of media control for retrieving playback information

For OITF devices that require retrieving the position and the duration parameter from the server for operational reasons, the OITF shall support the method GET_PARAMETER message for that purpose.

All OITF devices shall support the retrieval from the server of the scales parameter through the GET_PARAMETER message.

Any other parameter not supported by the cluster controller used in the GET_PARAMETER request shall be rejected by the cluster controller with an appropriate error code. An empty body shall be allowed for the RTSP keep-alive message.

If RTSP is used as a keep-alive, then the timeout for sending the request is based on the timeout parameter specified in the session header in the RTSP SETUP response. If timeout parameter is not specified then a default value of 60 s shall be used.

On receipt of the RTSP GET_PARAMETER request, the cluster controller shall forward the message to the chosen CDF.

The CDF shall respond with a 200 OK message with the requested values or empty in the case of a keep-alive message. The message shall be forwarded to the OITF.

8.2.1.1.3.3 Handling of beginning and end of stream

On receipt of the beginning-of-stream or end-of-stream indication from the CDF, the cluster controller may send an RTSP ANNOUNCE to the OITF with an indication that the beginning-of-stream or end-of-stream has been reached. The Notice header shall be included with the notice code value set to "2104 Start-of-Stream Reached" or "2101 End-of-Stream Reached".

NOTE The header and code is based on IETF DRAFT-STIEMERLING-RTSP-ANNOUNCE-01.

On receipt of the RTSP ANNOUNCE with an end-of-stream or beginning-of-stream indication, the OITF may take relevant actions to handle the event (e.g. terminating the session, rewinding the media stream). The OITF shall respond with a RTSP 200 OK.

8.2.1.1.3.4 Handling by the CDF for multiple PLAY messages in play state

The CDF shall not queue successive PLAY messages for processing while in a play state. An incoming new PLAY message shall result in an immediate termination by the CDF of the processing associated with a previous pending PLAY message if applicable.

8.2.1.1.3.5 Handling of CDF-initiated session termination

If the CDF detects an event that leads it to have to terminate a user session, the CDF shall send an RTSP ANNOUNCE to the cluster controller with an indication that the session shall be terminated. The Notice header shall be included with the notice code value set to "5402 Client session Terminated". The cluster controller shall forward the message to the OITF, and return an RTSP 200 OK to the CDF.

NOTE The header and code is based on IETF DRAFT-STIEMERLING-RTSP-ANNOUNCE-01.

On receipt of the RTSP ANNOUNCE with a client session terminate indication, the OITF shall return an RTSP 200 OK then proceed to terminate the session in accordance with 8.2.1.1.4.

8.2.1.1.3.6 Handling of cluster controller-initiated session termination

If the cluster controller detects an event that leads it to have to terminate a user session, the cluster controller shall send an RTSP ANNOUNCE to the OITF with an indication that the session shall be terminated. The Notice header shall be included with the notice code value set to “5402 client session terminated”.

NOTE The header and code is based on IETF DRAFT-STIEMERLING-RTSP-ANNOUNCE-01.

On receipt of the RTSP ANNOUNCE with a client session terminate indication, the OITF shall return an RTSP 200 OK then proceed to terminate the session in accordance with 8.2.1.1.4.

8.2.1.1.4 RTSP session teardown

To tear down a unicast session, the OITF shall use a RTSP TEARDOWN message and shall wait for a 200 OK response from the cluster controller. The cluster controller shall relay the RTSP TEARDOWN message to the CDF and relay the 200 OK message to the OITF.

8.2.1.1.5 Supported RTSP messages

The OITF acting as an RTSP Client and the cluster controller acting as an RTSP Server shall support at least the following messages: RTSP SETUP, RTSP TEARDOWN, RTSP DESCRIBE, RTSP PLAY, RTSP PAUSE, RTSP GET_PARAMETER, RTSP ANNOUNCE, and RTSP OPTIONS.

The CDF as an RTSP server shall support at least the following messages: RTSP SETUP, RTSP TEARDOWN, RTSP DESCRIBE, RTSP PLAY, RTSP PAUSE, RTSP OPTIONS, and RTSP GET_PARAMETER.

8.2.1.2 RTSP profile with SIP session management over UNIS -11 and NPI-10

8.2.1.2.1 General

The RTSP protocol shall be used on NPI-10 for unicast content streaming session setup and UNIS-11 and NPI-10 for unicast content streaming service delivery.

The OITF shall obtain the appropriate RTSP request URI, RTSP session ID, and the RTP media parameters, prior to content delivery from the assigned cluster controller.

If the OITF supports RTSP/RTCP monitoring, it shall also include the a=OIPF-QoS-Metrics line, the a=rtcp-xr line and the b=RR line prior to content delivery from the assigned CC, where:

- The a=OIPF-QoS-Metrics line includes information on the cumulative performance metrics the service provider requests from the client for that session (see 8.3).
- The a=rtcp-xr line includes information on the sample performance metrics the service provider requests from the client for that session (see 8.3).
- Finally, the b=RR line specifies the reporting bandwidth assigned to the OITF in bits per second, see Clause 2 of IETF RFC 3556. If this line is not retrieved via SIP OPTIONS, then the OITF shall use the RTP default of 5% of the stream bandwidth for RTCP reports.

The use of RTSP shall comply with IETF RFC 2326 with modifications defined by this specification.

Guidelines for specifying cumulative metrics to be conveyed using the RTSP headers defined in IEC 62766-4-2. Similarly to the RTCP case, “OIPF-BasicPerfMonCumulSubset1” is used in this specification as a placeholder and for illustrative purposes.

8.2.1.2.2 Missing SDP parameters retrieval

When the Cluster controller receives a SIP OPTIONS message to retrieve missing parameters, it shall send an RTSP DESCRIBE message to an appropriate CDF. The “Accept” header shall be set to “application/sdp”.

The CDF shall reply with a RTSP 200 OK message with the Content-type header set to “application/sdp”.

If the CDF replied with a redirection response, the cluster controller shall send a new RTSP DESCRIBE to the new CDF.

If the cluster controller failed to successfully complete the RTSP DESCRIBE transaction, it shall return an appropriate error message to the SIP OPTIONS message.

8.2.1.2.3 RTSP session setup

The OITF shall not use the RTSP SETUP message. The RTSP session setup is initiated with a SIP INVITE.

When receiving a COD session initiation SIP request from the CDN controller, the cluster controller shall choose the appropriate CDF and issue an RTSP SETUP message with the following parameters:

- the RTSP URI shall have a path that is compatible with the requested content indicated in the user part of the “To:” header of the SIP message;
- the Transport header shall contain a “Destination” sub header indicating the IP address of the OITF.

The CDF shall reply with an RTSP 200 OK message to the cluster controller.

- The message shall contain an RTSP session ID.
- The CSeq shall be set to the same value as in the RTSP SETUP request.

If the cluster controller receives an error message from the CDF, it may try another CDF. It may also reply with the appropriate SIP error message to the CDN controller (see 7.2.3.1.3).

The cluster controller shall issue a SETUP request for each media line (if the content is FEC protected).

If the new CDF sends a redirect response to the SETUP request, the cluster controller shall send a new SETUP request to the new CDF.

If the cluster controller failed to successfully complete the RTSP SETUP transaction, it shall return an appropriate error message to the SIP INVITE.

8.2.1.2.4 RTSP control for media delivery

8.2.1.2.4.1 Handling of media control for starting playback

On receiving a request from the user to start playback, the OITF shall follow the procedures defined in ETSI TS 183 063, 7.1.1.2 with the OITF acting as a UE and the cluster controller acting as an MCF.

8.2.1.2.4.2 Handling of media control for modifying playback

On receiving a request from the user to modify playback, the OITF shall follow the procedures defined in ETSI TS 183 063, 7.1.1.3 with the OITF acting as a UE and the cluster controller acting as an MCF.

On receipt of a RTSP PLAY or PAUSE request, the cluster controller shall forward the message to the chosen CDF.

The CDF shall respond with a 200 OK message. The contents of the 200 OK response shall be as follows:

- CSeq shall be set to the same value as that in the request.

8.2.1.2.4.3 Handling of media control for retrieving playback information

On receiving a request from the user to retrieve playback information, the OITF may send an RTSP GET_PARAMETER message. The OITF may retrieve the following information:

- Position:
The position in the media in seconds.
- Scales:
The allowed scales that can be used in the PLAY request. Syntax shall be a comma separated array of allowed scales.
- Duration:
The total duration in seconds of the media to be played.

Any other parameter not supported by the cluster controller used in the RTSP GET_PARAMETER request shall be rejected by the cluster controller with an appropriate error code. An empty body shall be allowed for the RTSP keep alive message.

On receipt of the RTSP GET_PARAMETER request, the cluster controller shall forward the message to the chosen CDF.

The CDF shall respond with a 200 OK message with the requested values. The message shall be forwarded to the OITF.

8.2.1.2.4.4 Handling of beginning and end of stream

On receipt of the beginning-of-stream or end-of-stream indication from the CDF, the cluster controller may send an RTSP ANNOUNCE to the OITF with an indication that the beginning-of-stream or end-of-stream has been reached. The "Notice" header shall be included with the notice code value set to "2104 Start-of-Stream-Reached" or "2101 End-of-Stream Reached".

NOTE The header and code is based on IETF draft-stiemerling-rtsp-announce-01. The RTSP version used in this specification is "1.0" and not "2.0" as in the examples in IETF draft-stiemerling-rtsp-announce-01.

On receipt of the RTSP ANNOUNCE with a beginning of stream or an end-of-stream indication, the OITF may take relevant actions to handle the end of stream event (e.g. terminating the session, rewinding the media stream, etc.). The OITF shall respond with a RTSP 200 OK.

8.2.1.2.4.5 Handling of CDF-initiated session termination

If the CDF detects an event that leads it to have to terminate a user session, the CDF shall send an RTSP ANNOUNCE to the cluster controller with an indication that the session shall be terminated. The Notice header shall be included with the notice code value set to "5402 Client session Terminated". The cluster controller shall perform the following:

- return an RTSP 200 OK to the CDF;
- proceed to tear down the RTSP session in accordance with subclause 8.2.1.1.4, and where the cluster controller plays the role of the OITF;
- proceed to tear down the IMS session with the CDNCF.

8.2.1.2.4.6 Handling of cluster controller-initiated session termination

If the cluster controller detects an event that leads it to have to terminate a user session, the cluster controller shall perform the following:

- proceed to tear down the RTSP session in accordance with subclause 8.2.1.1.4 and where the cluster controller plays the role of the OITF;
- proceed to tear down the IMS session with the CDNCF.

8.2.1.2.5 RTSP session teardown

The OITF shall not use the RTSP TEARDOWN message. The RTSP session teardown is initiated via SIP.

When the cluster controller receives a SIP BYE message to teardown a SIP unicast content streaming session, the cluster controller shall send a RTSP TEARDOWN message to the CDF. The CDF shall reply with a 200 OK.

The cluster controller shall issue an RTSP TEARDOWN request for each media line (if the content is FEC protected).

8.2.1.2.6 RTSP messages supported

The OITF acting as an RTSP Client shall support RTSP PLAY, RTSP PAUSE, RTSP GET_PARAMETER, RTSP ANNOUNCE, and RTSP OPTIONS.

The cluster controller acting as an RTSP Proxy and RTSP Client shall support at least the following messages: RTSP SETUP, RTSP TEARDOWN, RTSP DESCRIBE, RTSP PLAY, RTSP PAUSE, RTSP OPTIONS, and RTSP GET_PARAMETER.

The CDF acting as an RTSP server shall support at least the following messages: RTSP SETUP, RTSP TEARDOWN, RTSP DESCRIBE, RTSP PLAY, RTSP PAUSE, RTSP OPTIONS, and RTSP GET_PARAMETER.

8.2.2 Use of RTSP for forced play-out control RTSP for managed model over UNIS-11 and NPI-10

8.2.2.1 General

The RTSP/RTCP monitoring, RTSP session setup and teardown procedures are unchanged by Forced Play Out and shall follow the description in 8.2.1.

8.2.2.2 RTSP control for media deliver Forced play-out controlled by the cluster controller

On receiving a request from the user to control the playback (e.g. RTSP PLAY, PAUSE, etc), the OITF shall follow the procedures specified in 8.2.1.2.4.

On receiving a request from the user to control the playback (e.g. RTSP PLAY, PAUSE, etc), the cluster controller shall examine the request to see whether the playback operation is permitted based on the Forced Play Out Control policy. If the requested playback operation is forbidden by the policy, for example, the user tries to fast forward when an advertisement is showing, the cluster controller shall disable the request and respond with a RTSP 405 message. If the playback is permitted, the cluster controller shall forward the request to the selected CDF.

The CDF shall respond with a 200 OK message. The contents of the 200 OK response shall be as follows:

- CSeq shall be set to the same value as that in the request.

Handling of Media Control for Retrieving Playback Information is the same as specified in 8.2.1.2.4.

Handling of End of Stream is not affected by Forced Play Out and is the same as specified in 8.2.1.2.4.

8.2.3 Use of RTSP for network PVR (nPVR)

8.2.3.1 RTSP session setup

When performing recording, the cluster controller shall first retrieve from the incoming request the SDP of the scheduled content programme and then issue a DESCRIBE messages to the most appropriate CDF. The DESCRIBE message shall include the Accept header with the application/sdp content type. The CDF shall return a Content-Type header with application/sdp and the format of the body shall be according to IETF RFC 4566. The cluster controller shall then issue a SETUP request to CDF. The CDF then shall join the multicast group as defined in 9.2.3.

If the setup is successful, the CDF shall reply with a 200 OK message to the cluster controller.

- The message shall contain a RTSP Server port and a RTSP session ID.
- The CSeq shall be set to the same value as in the RTSP SETUP request.

After receiving the setup response, the cluster controller shall send RECORD message to the CDF. Then the CDF starts recording the content. When receiving error messages from the CDF, the cluster controller shall try another CDF.

8.2.3.2 Handling of end of recording

When finishing the recording, the CDF shall issue a RTSP ANNOUNCE message to the cluster controller to report the record result.

Within the message body, the CDF shall include a body associated with the appid "urn:oipf:npvr:report:2009". The parameters shall be set:

- Result: the result of the recording (i.e. Completed, Error).
- CRID: the identifier reference to the recorded content, for OITF to request later.
- Spare: the spare storage existing in the CDF.

8.2.3.3 RTSP session teardown

When the cluster controller receives a SIP BYE message to tear down a SIP nPVR session, the cluster controller shall issue a RTSP TEARDOWN message to the CDF. The CDF shall reply with a 200 OK.

8.2.4 Use of RTSP for personalized channel

8.2.4.1 RTSP session setup for PCh

When receiving a PCh session initiation SIP INVITE from the CDN controller, the cluster controller shall choose the appropriate CDF and shall issue an RTSP SETUP message with the following parameters:

- the RTSP URI shall contain a path that is compatible with the requested PCh item indicated in the user part of the "To:" header of the SIP message;
- the Transport header shall contain a "Destination" sub header indicating the IP address of the OITF if the content item is unicast content service; if the content item is multicast

content service, this header shall indicate the multicast address related to the content service.

On reception of RTSP SETUP message, the CDF shall reply with an RTSP 200 OK message to the cluster controller.

- The message shall contain an RTSP session ID.
- The CSeq shall be set to the same value as in the RTSP SETUP request.

If the requested PCh item is a multicast content service and the CDF doesn't cache the content, the CDF may fetch the content, e.g., join the associated multicast group and fetch the content using IGMPv3 as described in IETF RFC 3376.

8.2.4.2 RTSP PLAY for PCh content switch

When receiving a PCh content switch request SIP MESSAGE from the CDN controller, the cluster controller shall issue an RTSP PLAY message to the same CDF as selected in the PCh session initiation, with the following parameters:

- the RTSP URI shall contain a path that is compatible with the ContentId that is carried in the SIP INFO message body;
- the session ID shall be set to identically to that of the RTSP SETUP request;
- the Transport header shall contain a "Destination" sub header indicating the IP address of the OITF if the content item is unicast content service; if the content item is multicast content service, this header shall indicate the multicast address related to the content service.

On reception of RTSP PLAY message, the CDF shall reply with an RTSP 200 OK message to the cluster controller.

- The message shall contain the same session ID as the RTSP SETUP.
- The CSeq shall be set to the same value as in the RTSP SETUP request.

If the requested PCh item is multicast content service and the CDF doesn't cache the content, the CDF may fetch the content, e.g., join the associated multicast group and fetch the content using IGMPv3 as described in IETF RFC 3376.

8.2.5 Content streaming post-session transfer

The following procedure shall be undertaken by a transferee OITF to view the unicast content service, once it has successfully established the unicast content service session to handle the transfer:

- Step 1:** If the transferee OITF has the content bookmark, it shall send the RTSP PLAY to the CC/CDF and the bookmark shall be added in the Range header to indicate the content start time to be played. The CC/CDF shall send the related content through the content delivery channel established in the unicast content service session to the transferee OITF. The procedures in 8.2.1.2.4, and 8.2.1.2.5, for media control purposes can be reused and this procedure terminates.
- Step 2:** If the transferee OITF does not have the content bookmark, it can use the procedure defined in 6.3.9.5.1, to retrieve the content bookmark, and subsequently it can use the procedure defined in Step 1.

8.3 Protocols for service access and control

Performance monitoring over UNIT-18

QoS metrics can be classified as those that need to be reported regularly, i.e. 'sample metrics', and those that are typically required when the service ends, i.e., 'cumulative metrics'. Periodic RTCP reports are more appropriate for transport of sample metrics (see 10.3), while

on-demand or scheduled RTSP reports are especially suitable for transport of cumulative metrics. In general, an IPTV service might need a combination of both.

This subclause specifies the OPTIONAL RTSP mechanisms for performance monitoring of unicast content streaming services.

If the OITF supports QoS metrics and has been suitably configured to use them, then the unicast content streamign session initiation request over HNI-IGI interface shall include the selected (i.e. accepted by client) or modified (for re-negotiation) QoS metrics for either the session level or media level.

The QoS metrics negotiation shall start at the cluster controller (CC) on reception of a response to an RTSP DESCRIBE including metrics information embedded in the session description. The RTSP DESCRIBE at the CC is triggered by a SIP OPTIONS request for missing parameters from the CDN controller, as per subclause 8.2.1.2.3.

On receiving this SETUP request, the content delivery function (CDF) shall return the RTSP 200 OK response with the “accepted” QoS metrics (i.e. metrics and reporting values which are identical to the ones in the client’s request and accepted by the CDF) and the “re-negotiation” QoS metrics (i.e. metrics and reporting values which are not identical to the ones in the client’s request and modified for re-negotiation by the CDF). The echoing of the “accepted” QoS metrics is for re-acknowledging the client’s request.

The CDF may also reject the changes made by the client, i.e. reject the “re-negotiation” of QoS metrics. If the CDF rejects the changes, it shall either set new values and resend the modified metrics back to the client, or it shall ignore the “re-negotiation” metrics and not re-acknowledge them. Any QoS metric that has been acknowledged as “accepted” by the CDF Shall not be re-negotiated, i.e., it shall not be resent in the “OIPF-QoS-Metrics” header in the next RTSP request and shall not be re-acknowledged in the next RTSP response.

If the CDF does not approve the modifications made by the client, they should continue to re-negotiate. However, negotiations should not exceed 4 round trips, in order to minimize the potential delay of the negotiation process. The negotiation process may delay the start-up of the service and it may be avoided by carefully selecting the value of the *Metrics-Set* parameter in the service information. It shall be noted that each time the “QoS-Metrics” header field is sent in an RTSP request, it shall also be present in the response corresponding to that particular request. Otherwise, the receiver of the response shall assume that the other end does not support QoS metrics.

If there is no DESCRIBE request-response pair sending at the beginning of the RTSP session between the CC and the CDF, it means that the session description is received by other means. If such a description contains the “OIPF-QoS-Metrics” attribute, the negotiation starts at the CC with a SETUP request containing the “OIPF-QoS-Metrics” header.

If the session description does not contain the “OIPF-QoS-Metrics” attribute and the CDF would still like to check whether the client supports the QoS Protocol or not, the CDF shall include the “OIPF-QoS-Metrics” header containing the initial QoS metrics in the SETUP response. If the OITF sends the QoS metrics information in the next request (indicating that it supports the QoS Protocol), the negotiation shall continue until a mutual agreement is reached or the negotiation limit of 4 round-trips is reached.

To inform the client of the CDF’s desire to receive reports for the session, a new SDP attribute is specified to convey the QoS metrics. This attribute is defined inline with subclause 5.3.3.6 of 3GPP TS 26.234v750.

The ABNF definition (see IETF RFC 4234) is as follows:

```

OIPF-QoS-Metrics-Att = "a=" "OIPF-QoS-Metrics" ":" Measure-Spec *("," Measure-Spec) CRLF
Measure-Spec = "Metrics ";" Sending-rate [ ";" Measure-Range] *([ ";" Parameter-Ext])
Metrics = "cumul-metrics" "=" Metrics-Set / ("{" Metrics-Name *("|" Metrics-Name) "}")
Metrics-Name = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7e);VCHAR except ";", ",", "{", " or "
Metrics-Set = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7e);VCHAR except ";", ",", "{", " or "
Sending-Rate = "rate" "=" 1*DIGIT / "End"
Measure-Range = "range" ":" Ranges-Specifier
Parameter-Ext = "On"/"Off"/ (1*DIGIT ["." 1*DIGIT]) / (1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7c / 0x7e))
Ranges-Specifier = as defined in IETF RFC 2326
CRLF = %d13.10

```

This specification defines two new RTSP headers to negotiate and report the 'cumulative metrics' during session setup. These new RTSP headers follow the semantics of 3GPP TS 26.234v750, and are accordingly called OIPF-QoS-Metrics and OIPF-QoS-Feedback. They shall be used as follows:

- OIPF-QoS-Metrics shall be used for setting up and controlling the reporting of cumulative metrics, i.e. turn on/off reporting, negotiate the set of metrics, its frequency and the report range. This header can be sent in requests and responses of the RTSP Methods SETUP, SET_PARAMETER, OPTIONS (with session ID) and PLAY. The OITF should use the OPTIONS (with session ID) or SET_PARAMETER RTSP methods to turn off the QoS feedback.

The ABNF definition is as follows:

```

QoS-header = "OIPF-QoS-Metrics" ":" ("Off" / Measure-Spec *("," Measure-Spec)) CRLF
Measure-Spec = Stream-URL ";" ((Metrics ";" Sending-rate [ ";" Measure-Range] *([ ";" Parameter-Ext])) / "Off")
Stream-URL = "url" "=" "<">Rtsp-URL<">
Metrics = "cumul-metrics" "=" Metrics-Set / ("{" Metrics-Name *("|" Metrics-Name) "}")
Metrics-Set = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7e);VCHAR except ";", ",", "{", " or "
Metrics-Name = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7e);VCHAR except ";", ",", "{", " or "
Sending-Rate = "rate" "=" 1*DIGIT / "End"
Measure-Range = "range" ":" Ranges-Specifier
Parameter-Ext = "On"/"Off"/ (1*DIGIT ["." 1*DIGIT]) / (1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7c / 0x7e))
Ranges-Specifier = as defined in IETF RFC 2326
Rtsp-URL = as defined in IETF RFC 2326
CRLF = %d13.10

```

- OIPF-QoS-Feedback shall be used for sending (or requesting) the actual QoS metrics feedback to (or from) the CDF. This header can be sent in requests and responses of the following RTSP Methods: SET_PARAMETER (or GET_PARAMETER), or PAUSE Methods.

The ABNF definition follows:

```

Feedback-header = "OIPF-QoS-Feedback" ":" Feedback-Spec *("," Feedback-Spec) CRLF

```

```

Feedback-Spec = Stream-URL ";" 1*(";" Parameters) [";" Measure-Range]
Stream-URL = "url" "=" <">Rtsp-URL<">
Metrics-Set = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) /
0x7e);VCHAR except ";", "\", "{ or "}"
Parameters = Metrics-Name "=" "{" SP / (Measure * ("|" Measure)) "}"
Metrics-Name = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) /
0x7e);VCHAR except ";", "\", "{ or "}"
Rtsp-URL = as defined in IETF RFC 2326
Measure-Range = "range" ":" Ranges-Specifier
Ranges-Specifier = as defined in IETF RFC 2326
Measure = Value [SP Timestamp]
Value = (["-"]1*DIGIT [". " *DIGIT]) / 1*((0x21..0x2b) / (0x2d..0x3a) /
(0x3c..0x7a) /
VCHAR except ";", "\", "{ or "}"
Timestamp = NPT-Time
NPT-Time = as defined in IETF RFC 2326
CRLF = %d13.10

```

The OITF shall use the SET_PARAMETER method for cumulative QoS metrics reporting, using the OIPF-QoS-Feedback header defined in this document. However, in some cases, the RTSP method may also be used:

- When the client wants to pause the streaming flow, the QoS information should be embedded into a PAUSE method. The OITF should not send any QoS reports to the CDF when the media stream is paused, as no media is being exchanged.

The reporting CDF shall use the GET_PARAMETER method for retrieving cumulative QoS metrics from the OITF on-demand, using the OIPF-QoS-Feedback header defined in this specification.

The RTSP header and attribute definitions above are almost identical to those in 3GPP TS 26.234v750. The semantics shall comply with subclause 5.3.2.3 of 3GPP TS 26.234 except for the differences in naming and the following Open IPTV Forum specific changes:

- There is a possibility to simplify the request and reporting of a (potentially large) set of metrics by introducing a Metrics-Set attribute in the OIPF-QoS-Metrics and OIPF-QoS-Feedback headers. i.e., the Metrics-Set attribute may be used instead of listing each metric explicitly, as these would have empty values during the metrics negotiation phase anyway; this makes messages more compact when the number is large.

Note the commonalities between the definition of the RTSP header OIPF-QoS-Metrics and the SDP attribute above. Note also that the Stream-URL is not present in the SDP attribute; this is because the value of the Stream-URL is present in the session description and is thus known by the CC at the time of issuing the RTSP SETUP for each media line.

9 IGMP and multicast protocol

9.1 General

This clause defines the protocol for the use of IGMP and Multicast over the following reference points:

- UNIS-13;
- UNIS-7;
- UNIS-15;
- UINS-RMS;
- UNIS-6;

- UNIS-12;
- NPI-40.

9.2 Protocols for IPTV service functions

9.2.1 Multicast content streaming service on UNIS-13

9.2.1.1 General

The OITF shall support IGMPv3 as described in IETF RFC 3376. If the Transport Processing Function supports a lower IGMP version, the backward compatibility rules between the OITF and the Transport Processing Function shall conform to IETF RFC 3376 clause 7.

9.2.1.2 Procedure for multicast content streaming with SIP session management

The use of IGMP on UNIS-13 with SIP session management shall be as defined in ETSI TS 183 063, 8.1.2 and 8.2, where the OITF replaces the UE, the Transport Processing Function replaces the Transport Functions, the BTF replaces the ECF/EFF and the service discovery FE/IPTV Metadata control FE replaces the SSF.

9.2.1.3 Procedure for multicast content streaming

The use of IGMP on UNIS-13 without session initiation shall be as defined in subclause 9.2.1.2.

In the case there is no session initiation, the procedures described in 8.1.2 of ETSI TS 183 063 to set the Group/Multicast Address Records will not retrieve any channel or source address information from the session initiation step.

9.2.1.4 Procedure for multicast RET

The use of IGMP on UNIS-13 shall be defined as in 9.2.1.2 or as defined in 9.2.1.3.

In this context, the IP Multicast group and source address are not related to a “channel” but to an associated RET IP Multicast, as defined in Annex F of ETSI TS 102 034 and N.1.4.

9.2.2 Pay-per-view multicast content service with SIP session management

PPV multicast content service session initiation over UNIS-13 shall be same as multicast content streaming session initiation as described in 9.2.1.

9.2.3 nPVR Protocol over NPI-40

The controlling protocol of CDN for nPVR is IGMP. The use of IGMP on NPI-40 shall be as defined in 9.2.1.2. In this case, the CDF in the CDN acts as an OITF for the multicast content service.

When the CDF receives the request to record a program, it shall send an IGMP Join to the Transport Processing Functions.

9.2.4 Network-generated notification service

Whenever the network generated notification event was detected, the IPTV application shall send the notification message as well as the destination multicast group identifier to the MCDF over NPI-42. The destination multicast group identifier indicates the multicast group used for the network generated notification service which differs from the related multicast content service.

Upon receiving the notification message, the MCDF shall distribute the notification message to the multicast group identified by destination multicast group identifier, as defined in 10.2.3.

For the OITF, Network Generated Notification session initiation over UNIS-13 shall be the same as multicast content streaming session initiation as described in 9.2.1.

9.2.5 Emergency notification service

Upon receiving an emergency notification from Emergency services, the Notification services shall send the emergency notification message as well as the destination multicast group identifier to the MCDF over NPI-38.

Upon receiving the notification message, the MCDF shall distribute the emergency notification message to the multicast group identified by destination multicast group identifier as defined in 10.2.3.

Emergency notifications are delivered to an OITF over UNIS-13, when the OITF joins the corresponding multicast group with the access parameters derived from SD&S as defined in IEC 62776-3.

9.3 Protocols for service access and control functions

9.3.1 Service discovery and Content Selection

9.3.1.1 Protocol over UNIS-15 and UNIS-7

DVB SD&S Transport Protocol (DVBSTP) specified in subclause 5.4.1 of ETSI TS 102 034 shall be used to transport the service discovery and Content Metadata related information over multicast.

9.3.1.2 Protocol over UNIS-13

The use of IGMP on UNIS-13 for service discovery and Content Selection shall be as defined in ETSI TS 183 063, 8.1.1 where the OITF replaces the OIPF, the service provider discovery FE replaces the SDF and the service discovery FE/Metadata control FE replaces SSF.

9.3.2 Remote management – protocol over UNI-RMS

The UNI-RMS provides the functions for the remote management of the OITF devices. This interface shall be based on Broadband Forum TR-069 Remote Management Framework TR-069.

9.4 Protocols for system infrastructure functions – interactive application delivery – protocol over UNIS-6 and UNIS-12

The use of multicast is an OPTIONAL feature for reducing network and server traffic when DAE and PAE applications are delivered to the OITF and AG respectively.

FLUTE IETF RFC 3926 shall be used when DAE and PAE applications are delivered through multicast. The FDT-Instance XML Schema defined in IETF RFC 3926 is extended with two additional attributes: Tags and Priority. The Tags attribute contains a list of tags that the content is associated with. The optional Priority attribute is used by the OITF to determine which content items can be discarded when there is a need to recover memory. The Priority attribute takes values between 1 and 10, with 10 being the highest priority.

The detailed schema extensions are described in Annex U.

The Content-Location element in the FDT-Instance shall be the same URI which is used to fetch the object with unicast, e.g., `http://server/DAE_pictures/background.jpg`. If the OITF has stored an object, it shall not download the object again using unicast unless it has expired.

If certain parts of the file are lost, the OITF may fetch the missing parts using HTTP with range headers.

10 RTP/RTCP

10.1 General

This clause defines the protocol for the use of RTP and RTCP over the following reference points:

- UNIT-17;
- UNIT-18;
- NPI-41.

10.2 Protocols for IPTV service functions

10.2.1 Multicast content streaming service

10.2.1.1 General

Scheduled content may be delivered as a multicast stream over UNIT-17.

10.2.1.2 Protocol over UNIT-17

The use of RTP on this reference point shall comply with 10.2.

10.2.2 Unicast content streaming service

10.2.2.1 General

Streamed Scheduled Content or Content on Demand may be delivered as a unicast stream over UNIT-17. RTP and HTTP may be used. This subclause specifies the use of RTP.

10.2.2.2 Protocol over UNIT-17

RTP shall be used on this reference point and shall comply with 10.2.

The use of RTCP shall comply with 10.3.

10.2.3 nPVR

Protocol over NPI-41

This interface is used for multicast delivery from the Transport Processing Functions to the CDF in the CDN for the purposes of nPVR. RTP shall be used on NPI-41 and shall comply with subclause 10.2.

10.3 Service access and control – Performance monitoring over UNIT-18

This subclause specifies the optional setup and reporting of sample metrics for unicast content streaming services.

The setup is initiated when the SDP session parameters are retrieved. A CDF requesting sample metrics reports via RTCP shall include the performance reporting information in the retrieved SDP information using the `a=rtcp-xr` SDP attribute as defined below.

RTCP shall be used as per 7.1.1.1 of ETSI TS 102 034 with the following additions:

- RTCP Receiver Reports defined by IETF RFC 3550 shall be used to include RTCP XR extended reports following the rules of IETF RFC 3611 and as defined further below.
- The RTP default value for RTCP bandwidth of 5% may be overridden by using the SDP bandwidth modifiers as specified in IETF RFC 3556. See 8.2.1.2 for details.

Unlike the case of cumulative metrics, the sample metrics are not present in the OIPF-QoS-Metrics RTSP header as there is already a framework for reporting metrics using RTCP, namely the RTCP XR extended report as per IETF RFC 3611. A consequence of this is that the sample metrics cannot be negotiated, as RTSP cannot negotiate parameters present in the SDP.

The SDP attributes to define the required sample metrics in accordance with 5.1 of IETF RFC 3611 are as follows:

```
rtcp-xr-attr = "a=" "rtcp-xr" ":" [xr-format *(SP xr-format)] CRLF
xr-format = default-OIPF-xr-report / new-OIPF-xr-report
default-OIPF-xr-report = "OIPF-BasicPerfMonSampleSubset1"
new-OIPF-xr-report = non-ws-string; non-white-space string
non-ws-string = 1*(%x21-FF)
CRLF = %d13.10
```

An OITF using RTCP reporting shall support the extended report blocks defined in future versions of this specification. Guidelines for specifying RTCP XR Extended Report containing sample metrics are provided in IEC 62766-4-2. *OIPF-BasicPerfMonSampleSubset1* is used in this specification as a formal placeholder and for illustrative purposes.

The OITF may support other RTCP XR extended reports as defined in IETF RFC 3611 (e.g., pkt-loss-rle, pkt-dup-rle, pkt-rcpt-times) but these are not required for the performance monitoring to work correctly.

10.4 Application-layer forward error correction

This subclause specifies the protocol for application Layer FEC (AL-FEC) protection of streaming media for multicast content services carried over RTP transport.

The application Layer FEC conforms to Annex E of ETSI TS 102 034. Only the base layer of DVB-IP AL-FEC is supported in this specification, the enhancement layer support is out of scope.

DVB AL-FEC base layer is signalled in DVB SD&S, as defined in 5.2.6.2 of ETSI TS 102 034.

10.5 Application-layer retransmission (RET)

10.5.1 General

This subclause specifies the RTP retransmission solution both for multicast content services and unicast content services transported over RTP, to provide for packet loss repair service. The solution shall be conformant to the application layer retransmission (RET) solution as specified in Annex F of ETSI TS 102 034.

10.5.2 Protocol over UNIT-17

The unicast RTP retransmission data associated with multicast content service or unicast content service is transferred over the UNIT-17, in conformance with ETSI TS 102 034.

For unicast content services, RTP retransmission packets shall be SSRC multiplexed with the original RTP packets.

For multicast content services, in addition to unicast RTP retransmission data, there may also be multicast RTP retransmission data and related multicast RTCP retransmission information (such as RTCP FeedForward NACK message) transferred over UNIT-17, as specified in ETSI TS 102 034.

10.5.3 Protocol over UNIT-18

Unicast RTCP shall be used on this reference point for RET requesting -by means of the RTCP FeedBack NACK message- and reporting for both multicast content service and unicast content service and shall comply with the Annex F of ETSI TS 102 034.

10.6 Fast channel change (FCC)

10.6.1 General

This subclause specifies the fast channel change solution for multicast scheduled content services transported over RTP. The solution shall be conformant to the server-based fast channel change solution as specified in Annex I of ETSI TS 102 034.

10.6.2 Protocol over UNIT-17

The unicast Fast Channel Change RTP burst data is transferred over the UNIT-17, in conformance with Annex I of ETSI TS 102 034.

10.6.3 Protocol over UNIT-18

RTCP shall be used on this reference point and shall comply with Annex I of ETSI TS 102 034.

10.7 Protocols for multimedia telephony services – protocol over UNIT-17

10.7.1 General

The use of RTP on this reference point shall comply with IETF RFC 3550 and IETF RFC 3551.

10.7.2 RTP payload formats for video streams

The RTP payload format defined in IETF RFC 3984 shall be used for the transport of H.264 video streams according to the video media type AVC_VDC as defined in IEC 62766-2-1. The following restrictions on the usage of IETF RFC 3984 shall apply:

- the interleaved packetization mode shall not be used,
- the single NAL unit and the non-interleaved packetization modes shall both be supported.

The RTP payload format defined in IETF RFC 3016 shall be used for the transport of MPEG-4 Part-2 video streams according to the video media type MP4V as defined in IEC 62766-2-1.

The RTP payload format defined in IETF RFC 4629 shall be used for the transport of H.263 video streams according to the video media type H263 as defined in IEC 62766-2-1.

10.7.3 RTP payload formats for audio streams

The RTP payload formats defined in IETF RFC 3551 shall be used for the transport of G.711 and G.722 speech audio streams according to the audio media type G711 and G722 as defined in IEC 63766-2-1.

The RTP payload format defined in IETF RFC 4867 and further specified in 7.4.2 of 3GPP TS 26.114 shall be used for the transport of AMR-NB and AMR-WB speech audio streams according to the audio media types AMR and AMRWB as defined in IEC 63766-2-1.

The RTP payload format defined in IETF RFC 4749 shall be used for the transport of G.729.1 speech audio streams according to the audio media type G7291 as defined in IEC 63766-2-1.

The RTP payload format defined in IETF RFC 5404 shall be used for the transport of G.719 audio streams according to the audio media type G719 as defined in IEC 63766-2-1.

The RTP payload format defined in IETF RFC 3640 shall be used for the transport of MPEG-4 audio streams according to the audio media types AACLD and AACELD as defined in IEC 63766-2-1.

11 UPnP protocols for system infrastructure functions – UPnP discovery

11.1 General

OITFs that do not support native HNI-IGI do not support UPnP-based IG discovery.

11.2 Procedure for IG discovery

11.2.1 Discovery sequence

When an OITF powers up, the OITF shall automatically discover the IG using the UPnP discovery Mechanism defined by the UPnP Device Architecture. A summary of the steps are as follows:

- Step 1:** An OITF sends the UPnP search request with the search target (urn:oipf-org:device:ig:1) to the specific multicast IP/Port address (239.255.255.250:1900).
- Step 2:** When an IG receives the search request, the IG sends the response message with its UPnP device description location (URL, e.g. http://IGAddress/Description.xml) to the requester's IP address by HTTP-U protocol.
- Step 3:** The OITF sends the HTTP GET request for retrieving the UPnP device description from the location (e.g. http://IGAddress/Description.xml).
- Step 4:** The IG sends the response with its UPnP device description, which holds the IG description.

11.2.2 urn:oipf-org:device:ig:1 device definitions

This subclause defines the urn:oipf-org:device:ig:1 deviceType as shown in Table 144.

Table 144 – urn:oipf-org:device:ig:1 deviceType

deviceType	Root	R/O	serviceType	R/O	serviceID
urn:oipf-org:device:ig:1	Root or Embedded	Required	see below	n/a	n/a

As described above, the urn:oipf-org:device:ig:1 deviceType does not have any specific definition for the serviceType it supports. It may have services of any serviceType.

11.2.3 IG description

To interact with the IG, the OITF shall know the IG URI and possibly a set of supported methods. The device element of the device description document for the urn:oipf-org:device:ig:1 deviceType shall contain this information, IG description, which is described as an XML fragment. The XML schema for the IG description is depicted in Figure 11.

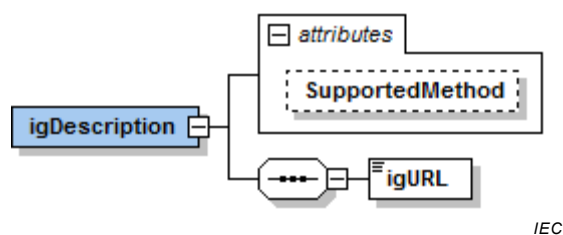


Figure 11 – IG description XML schema

The XML schema for the IG description is defined as follows:

```
<xs:schema targetNamespace="urn:oipf-org:device:ig:1"
  xmlns:tns="urn:oipf-org:device:ig:1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="igDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="igURL" type="xs:anyURI" />
      </xs:sequence>
      <xs:attribute name="SupportedMethod"
        type="tns:Hexadecimal16bit" use="optional" />
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="Hexadecimal16bit">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9a-fA-F]{1,4}" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Table 145 lists and describes the elements and attributes of the IG description schema.

Table 145 – igDescription elements and attributes

Element/Attribute Name	Description
igDescription	The root element of the IG Description XML fragment
@SupportedMethod	Hexadecimal16bit to describe the methods supported by the IG over the HNI-IGI interface (Subclause 6.6), which correspond to SIP methods and other functionality. Each bit represents a supported method as follows: 0001: REGISTER 0002: INVITE 0004: BYE 0008: CANCEL 0010: SUBSCRIBE 0020: NOTIFY 0040: PUBLISH 0080: MESSAGE 0100: OPTIONS 0200: GBA authentication 0400: HTTP Digest authentication (0800~8000: Upper 5 bits are currently not assigned but these bits could be used for other methods defined later) The value of this attribute is the result of a 16 bit OR operation with representative values of supported methods. If the SupportedMethod attribute is not described, the IG shall support all methods.
igURL	IG URL for the HNI-IGI (refer to 6.6). It may be relative to base URL (URLBase of uPnP device description if it exists or the URL from which the device description was retrieved). It shall not exceed 256 bytes in URI-escapes UTF-8 encoded form.

Note that GBA authentication can be achieved using either the GBA authentication using IMS Gateway procedure, specified in IEC 62766-7, 5.4.5 or the more general procedure, HTTP Digest authentication, using IMS Gateway in IEC 62766-7, 5.4.4. The latter, more general, procedure allows the use of different authentication mechanism in a way that is transparent to the OITF, including possible future authentication mechanisms, and should preferably be used. It is expected that GBA authentication using IMS Gateway procedure will be deprecated and removed in future versions of this specification.

The namespace of this fragment is “urn:oipf-org:device:ig:1”.

Example:

```
<ig:igDescription xmlns:ig="urn:oipf-org:device:ig:1"
SupportedMethod="01ff">
<ig:igURL>
  http://192.168.0.2/IG/
</ig:igURL>
</ig:igDescription>
```

11.3 Procedure for AG discovery

11.3.1 Discovery sequence

When an OITF powers up, the OITF shall automatically discover the AG using the UPnP discovery Mechanism defined by the UPnP Device Architecture. A summary of the steps are as follows:

- Step 1:** The OITF sends the UPnP search request with the search target (urn:oipf-org:device:ag:1) to the specific multicast IP/Port address (239.255.255.250:1900).
- Step 2:** When an AG receives the search request, the AG sends the response message with its UPnP device description location (URL, e.g. http://AGAddress/Description.xml) to the requester's IP address by HTTP-U protocol.
- Step 3:** The OITF sends the HTTP GET request for retrieving the UPnP device description from the location (e.g. http://AGAddress/Description.xml).
- Step 4:** The AG sends the response with its UPnP device description, which holds the AG description.

11.3.2 urn:oipf-org:device:ag:1 device definitions

This subclause defines the urn:oipf-org:device:ag:1 deviceType as shown in Table 146.

Table 146 – urn:oipf-org:device:ag:1 deviceType

deviceType	Root	R/O	serviceType	R/O	serviceID
urn:oipf-org:device:ag:1	Root or Embedded	Required	see below	n/a	n/a

As described above, the urn:oipf-org:device:ag:1 deviceType does not have any specific definition for serviceType it supports. It may have services of any serviceType.

11.3.3 AG description

The OITF may interact with the AG in one of two ways.

The applications running in the AG may provide remote UI as defined by the XML UI listing in CEA-2014-A.

The applications running in the AG may provide non-UI based services, for example listening for XML HTTP requests from DAE applications. No specific methods or services are defined.

To interact with the AG, the OITF shall know the AG URL and possibly a set of supported methods. The device element of the device description document for the urn:oipf-org:device:ag:1 deviceType shall contain this information, AG description, which is described as an XML fragment. The XML schema for the AG description is as follows:

Table 147 lists and describes the elements and attributes of the AG description schema.

```
<xs:schema targetNamespace="urn:oipf-org:device:ag:1"
  xmlns:tns="urn:oipf-org:device:ag:1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="agDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="agDefaultURL" type="xs:anyURI" />
        <xs:element name="agUIServerURL" type="xs:anyURI" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Table 147 – agDescription elements and attributes

Element/Attribute Name	Description
agDescription	The root element of the AG Description XML fragment
agDefaultURL	URL describing default address of AG, e.g. http://10.1.1.2
agUIServerURL	URL or URLs for remote UI provided by applications running on the AG

The namespace of this fragment is “urn:oipf-org:device:ag:1”.

11.4 Procedure for CSPG-DTCP discovery

11.4.1 Discovery sequence

When an OITF powers up, the OITF shall automatically discover the CSPG-DTCP using the UPnP discovery Mechanism defined by the UPnP Device Architecture. A summary of the steps are as follows:

- Step 1:** An OITF sends the UPnP search request with the search target (urn:oipf-org:device:cspg-dtcp:1) to the specific multicast IP/Port address (239.255.255.250:1900).
- Step 2:** When a CSPG-DTCP receives the search request, the CSPG-DTCP sends the response message with its UPnP device description location (URL, e.g. http://CSPGDTCPAddress/Description.xml) to the requester's IP address by HTTP-U protocol.
- Step 3:** The OITF sends the HTTP GET request for retrieving UPnP device description with the location (e.g. http://CSPGDTCPAddress/Description.xml).
- Step 4:** The CSPG-DTCP sends the response with its UPnP device description which holds the CSPG-DTCP description.

11.4.2 urn:oipf-org:device:cspg-dtcp:1 device definitions

This subclause defines the urn:oipf-org:device:cspg-dtcp:1 deviceType as shown in Table 148.

Table 148 – urn:oipf-org:device:cspg-dtcp:1 deviceType

deviceType	Root	R/O	serviceType	R/O	serviceID
urn:oipf-org:device:cspg-dtcp:1	Root or Embedded	Required	see below	n/a	n/a

As described above, the urn:oipf-org:device:cspg-dtcp:1 deviceType does not have any specific definition for serviceType it has. It may have services of any serviceType.

11.4.3 CSPG-DTCP description

To interact with the CSPG-DTCP, the OITF shall know which DRM system is supported, the port number used for DTCP key exchange and on which port the different content access protocols are proxied. The device element of the urn:oipf-org:device:cspg-dtcp:1 deviceType shall contain this information, and is described based on the following XML schema:

```
<xs:schema targetNamespace="urn:oipf-org:device:cspg-dtcp:1"
  xmlns:tns="urn:oipf-org:device:cspg-dtcp:1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="cspgdtcpDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DtcpPort" type="xs:integer" />
        <xs:element name="HttpProxyPort" type="xs:integer" />
        <xs:element name="RtspProxyPort" type="xs:integer" />
        <xs:element name="DRMSystemID" type="xs:anyURI"
          maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Table 149 lists and describes the elements and attributes of the CSPG-DTCP description schema.

Table 149 – cspgdtcpDescription elements and attributes

Element/Attribute Name	Description
cspgdtcpDescription	The root element of CSPG-DTCP Description XML fragment
DtcpPort	TCP port number for DTCP-AKE
HttpProxyPort	TCP port number for HTTP proxy in CSPG-DTCP
RtspProxyPort	TCP port number for RTSP proxy in CSPG-DTCP
DRMSystemID	Supported DRM system. The format of this attribute complies with DRMSystemId as defined in DRMControlInformation extension in Part 3.

The namespace of this fragment is "urn:oipf-org:device:cspg-dtcp:1".

Example:

```
<cg:cspgdtcpDescription xmlns:cg="urn:oipf-org:device:cspg-dtcp:1">
  <cg:DtcpPort>12345</cg:DtcpPort>
  <cg:HttpProxyPort>12346</cg:HttpProxyPort>
  <cg:RtspProxyPort>12347</cg:RtspProxyPort>
  <cg:DRMSystemID>urn:dvb:casystemid:12348</cg:DRMSystemID>
  <cg:DRMSystemID>urn:dvb:casystemid:12349</cg:DRMSystemID>
</cg:cspgdtcpDescription>
```

11.5 Procedure for OITF discovery

11.5.1 Discovery sequence

When an OITF powers up, the IG shall automatically discover the OITF using the UPnP discovery Mechanism defined by the UPnP Device Architecture. A summary of the steps are as follows:

- Step 1:** The IG sends the UPnP search request with the search target (urn:oipf-org:device:oif:1) to the specific multicast IP/Port address (239.255.255.250:1900)
- Step 2:** When an OITF receives the search request, the OITF sends the response message with its UPnP device description location (URL, e.g. http://OITFAddress/Description.xml) to the requester's IP address
- Step 3:** The IG sends the HTTP GET request for retrieving the UPnP device description from the location (e.g. http://OITFAddress/Description.xml)
- Step 4:** The OITF sends the response with its UPnP device description, which holds the OITF description.

11.5.2 OITF description

This subclause defines the urn:oipf-org:device:oif:1 deviceType as shown in Table 150.

The OITF can act as either a Media Renderer or a Media Server, and these two kinds of devices are defined in UPnP DCP (Device Control Protocol) related specifications (ISO/IEC 29341-4-13). Detailed descriptions and schemas of the serviceTypes are defined in these specifications.

Table 150 – urn:oipf-org:device:oif:1 deviceType

deviceType	Root	R/O	serviceType	R/O	serviceID
MediaRenderer:1	Root or Embedded	Required	RenderingControl:1.0	Required	RenderingControl
			ConnectionManager:1.0	Required	ConnectionManager
			AVTransport:1.0	optional	AVTransport
MediaServer:1	Root or Embedded	Required	ContentDirectory:1.0	Required	ContentDirectory
			ConnectionManager:1.0	Required	ConnectionManager
			AVTransport:1.0	optional	AVTransport

The OITF can also act as a digital media Player, but this device type is not discoverable so no device description is needed.

12 DLNA function

DLNA function is an optional gateway function which serves IPTV content to other DLNA devices in a consumer network. It converts the IPTV protocols, such as metadata access and media delivery protocols, to DLNA protocols. It may also convert media format and content protection schemes, if necessary. The interface between the DLNA function of the OITF and DLNA devices shall be compliant with IEC 62481-2.

13 DHCP

13.1 General

This clause defines the protocol for the use of DHCP over the following reference points:

- UNIT-16

13.2 Protocols for system infrastructure functions – network attachment

13.2.1 General

Network attachment provides IP addresses and configuration information to elements in the Consumer Domain prior to any other action regarding IPTV services. The provision and management of IP addresses has two main aspects.

IP address management within the consumer network: Deals with the attachment of the IG, AG and OITF to the WAN Gateway. The WAN Gateway shall act as a DHCP Server and a NAT. This type of attachment allows the IG, AG and OITF to communicate with each other within the consumer network. This allows the OITF to send and receive messages to and from the Internet.

IP address management for communication with the provider network: Two cases are supported:

- **MANDATORY:** The WAN Gateway translates the in-home IP address to an IP address recognizable to the provider's addressing plan. In this case a NAT shall be supported.
- **OPTIONAL:** The WAN Gateway assigns an IP address to the IG, AG and OITF from the managed network's IP addressing pool. In this case, NAT is not required.

Configuration information (e.g. DNS server) shall be provided to the OITF, AG and IG.

At power up all devices in the consumer network (OITF, IG, AG and other devices) shall request an IP address and network configuration parameters from the WAN gateway using DHCP.

13.2.2 DHCP option usage

13.2.2.1 Common options

The following is the minimum set of DHCP options defined in IETF RFC 3442 and IETF RFC 2132 that shall be used by the OITF, IG and AG when requesting DHCP configuration information from the WAN Gateway:

- option 1: Subnet Mask
- option 6: DNS
- option 61: Client Identifier. In this specification, the DHCP Client Identifier used in OITF devices is the deviceID, defined as follows:
 - deviceID – Identifies the device. It shall be unique within the home network and shall not change between restarts. The deviceID shall be the SHA-1 hash of the MAC address of the interface used to connect to the IPTV service as bytes concatenated with the domain name received via DHCP option 15 in ASCII characters.
 - deviceID = SHA-1(X)
where: X = (MAC address as bytes) + (domain name in ASCII characters) and the '+' denotes the concatenation operation. SHA-1 shall be used as specified in FIPS PUB 180-1. The domain name shall be set to the domain name received via DHCP option 15 (see 13.2.2.2).

For the IG and AG to support TR-069 based remote management, the following shall be supported:

- option 43: Vendor Specific Information is used to retrieve the Remote Management Server IP address or fully qualified domain name (FQDN). This information shall be provided by the DHCP server.

- option 60: Vendor Class identifier is used to indicate to the DHCP server that it is compliant with the Broadband Forum TR-069 specification. The vendor-class identifier shall be set to "IG_IPTV" or "AG_IPTV" by the IG or AG, respectively.

If the OITF supports TR-069 based remote management, the following shall be supported:

- option 43: Vendor Specific Information is used to retrieve the Remote Management Server IP address or fully qualified domain name (FQDN). This information shall be provided by the DHCP server.
- option 60: Vendor Class identifier is used to indicate to the DHCP server that it is compliant with the Broadband Forum TR-069 specification. The vendor-class identifier shall be set to "OITF_IPTV" by the OITF.

For managed networks relying on IMS, the following shall be supported:

- option 120: The address of the P-CSCF as per subclause 7.1.1 of ETSI TS 183 019.

13.2.2.2 Option 15

Option 15 shall be used by the OITF to request the domain name used to generate the device identity (deviceID) as per subclause 7.2.4.2.1.

13.2.2.3 Option 124/125

The OITF shall send a Vendor-Identifying Vendor Class option 124 as specified in IETF RFC 3925 when it requests a DHCP lease from the WAN Gateway. The option is specified with an enterprise-number of 37339 and the vendor-class-data identifier as "OITF_IPTV".

The DHCP server delivers the service provider discovery entry point via Vendor-Identifying Vendor-Specific Information DHCP option 125 as defined in IETF RFC 3925, with the enterprise-number and vendor-class-data identified from option 124.

In this specification, the service provider discovery entry points used are either:

- The FQDN/IP address of the IG, as per Clause E.1, or
- The FQDN/IP address of the service provider discovery functional entity (SP discovery FE), as per 7.2.4.1.

Format of DHCP payload:

The format of the vendor-specific binary buffer containing addresses returned by the DHCP server is a list of sub-options starting with sub-option number (one byte), sub-option length (one byte) and sub-option value (list of bytes).

The following vendor-specific sub-options are defined:

- Sub-option: IPTV-ENTRYPOINT: Code=0x01. This option carries either an IP Address or a fully-qualified domain name, as determined by a one byte "enc" field is used to indicate the type of encoding.
 - If the "enc" field has a value of 0x01, then this indicates an IP Address. The "enc" field is followed by 4 bytes corresponding to the IP Address. This value is used for the service provider discovery Entry point function.
 - If the "enc" field has a value of 0x02, then this indicates a FQDN (Fully-Qualified Domain Name). This value is used for the service provider discovery Entry point function.
 - The code of 0xFF is used to indicate end of the buffer.

- Sub-option: FCC/RET server location: Code=0x02. This option carries either a comma delimited list of IP Addresses or a comma delimited list of fully-qualified domain names, as determined by a one byte “enc” field is used to indicate the type of encoding.
 - If the “enc” field has a value of 0x01, then this indicates a list of comma-delimited IP Addresses.
 - If the “enc” field has a value of 0x02, then this indicates a list of comma-delimited FQDNs (Fully-Qualified Domain Name).
 - The code of 0xFF is used to indicate end of the buffer.
 - In either case, the servers shall be in the order of priority from first to last server to connect to.
- In conformance with Annex I of ETSI TS 102 034, the values for the FCC/RET server location(s) retrieved with DHCP are overruled by the SD&S elements “/Broadcastdiscovery/serviceList/SingleService/serviceLocation/IPMulticastAddress/Server-based LMB Enhancement service/RTCPReporting@DestinationAddress” providing the address of the FCC/RET server per each channel, when present.

NOTE This document does not define how to configure the DHCP server on the WAN gateway for FCC/RET server location.

14 UDP for IPTV service functions

14.1 Multicast content streaming service – protocol over UNIT-17

When MPEG2-TS are encapsulated directly in UDP (User Datagram Protocol), the encapsulation shall conform to ETSI TS 102 034, 7.1.2.

The use of RTP or direct UDP encapsulation shall be signalled by service discovery and Selection for multicast streaming. The use of RTP or direct UDP encapsulation shall be signalled in the SDP for unicast streaming.

In addition, it should be possible for an OITF to detect the usage of RTP or direct UDP encapsulation by looking for the value 0x47 in the first byte after the UDP header. In the case of Direct UDP encapsulation, this is the first byte of a 188 byte MPEG2-TS packet which always have the value 0x47 (synchronization byte of transport stream header – for any other value, RTP encapsulation is used).

14.2 Unicast content streaming service – Protocol over UNIT-17

The use of UDP on this reference point shall be as specified in 14.1.

14.3 nPVR – protocol over NPI-41

The use of UDP on this reference point shall be as specified in 14.1.

15 FLUTE

15.1 General

This clause defines the protocol for the use of FLUTE over the following reference points:

- UNIT-19

15.2 Protocols for IPTV service Functions

15.2.1 Emergency notification

15.2.1.1 General

Emergency notification is delivered in a multicast session over UNIT-19.

15.2.1.2 Protocol over UNIT-19

The use of FLUTE shall comply with IETF RFC 3926. When transported, an emergency notification message shall be carried in the FLUTE session as a TO (transport object), while FDT (File Delivery Table) shall be used to describe the notification message. The schema of FDT Instance shall conform to IETF RFC 3926 with the extension defined in subclause 9.4. The FDT shall comply with the schema defined in Annex P.

15.2.2 Network-generated notification

15.2.2.1 General

Network generated notification is delivered via a multicast session over UNIT-19. When the network generated notification is not tightly synchronized with a scheduled multicast content service, FLUTE shall be used.

15.2.2.2 Protocol over UNIT-19

The use of FLUTE shall comply with IETF RFC 3926. When transported, a network generated notification message shall be carried in the FLUTE session as a TO (transport object), while FDT (File Delivery Table) shall be used to describe the notification message. The schema of FDT Instance shall conform to IETF RFC 3926 with the extension defined in 9.4. The FDT shall comply with the schema defined in Annex P.

16 Diameter

16.1 General

This clause defines the protocol for the use of Diameter protocol IETF RFC 3588 over the following reference points:

- NPI-11

16.2 Protocols for IPTV service functions – Purchase of digital media – Protocol over NPI-11

The use of Diameter protocol shall comply with 3GPP TS 32.299 and IETF RFC 3588. Two messages will be used, Accounting-request (ACR) message and Accounting-Answer (ACA) message. The ACR message format is defined according to IETF RFC 3588 (see 6.2.2 of 3GPP TS 32.299 to get more information about ACR). The ACA message format is defined according to IETF RFC 3588 (see 6.2.3 of 3GPP TS 32.299 to get more information about ACA):

- Step 1:** The IPTV control FE sends ACR message to Charging FE to charge for the selected digital media. The ACR message is defined in 6.2.2 of 3GPP TS 32.299.
- Step 2:** The Charging FE returns ACA message to IPTV control FE with purchase result. The ACA message is defined in 6.2.3 of 3GPP TS 32.299.

Annex A (informative)

Example messages

A.1 IPTV service functions message examples – Unicast content streaming session setup with SIP session management

Figure A.1 shows a COD session setup sequence.

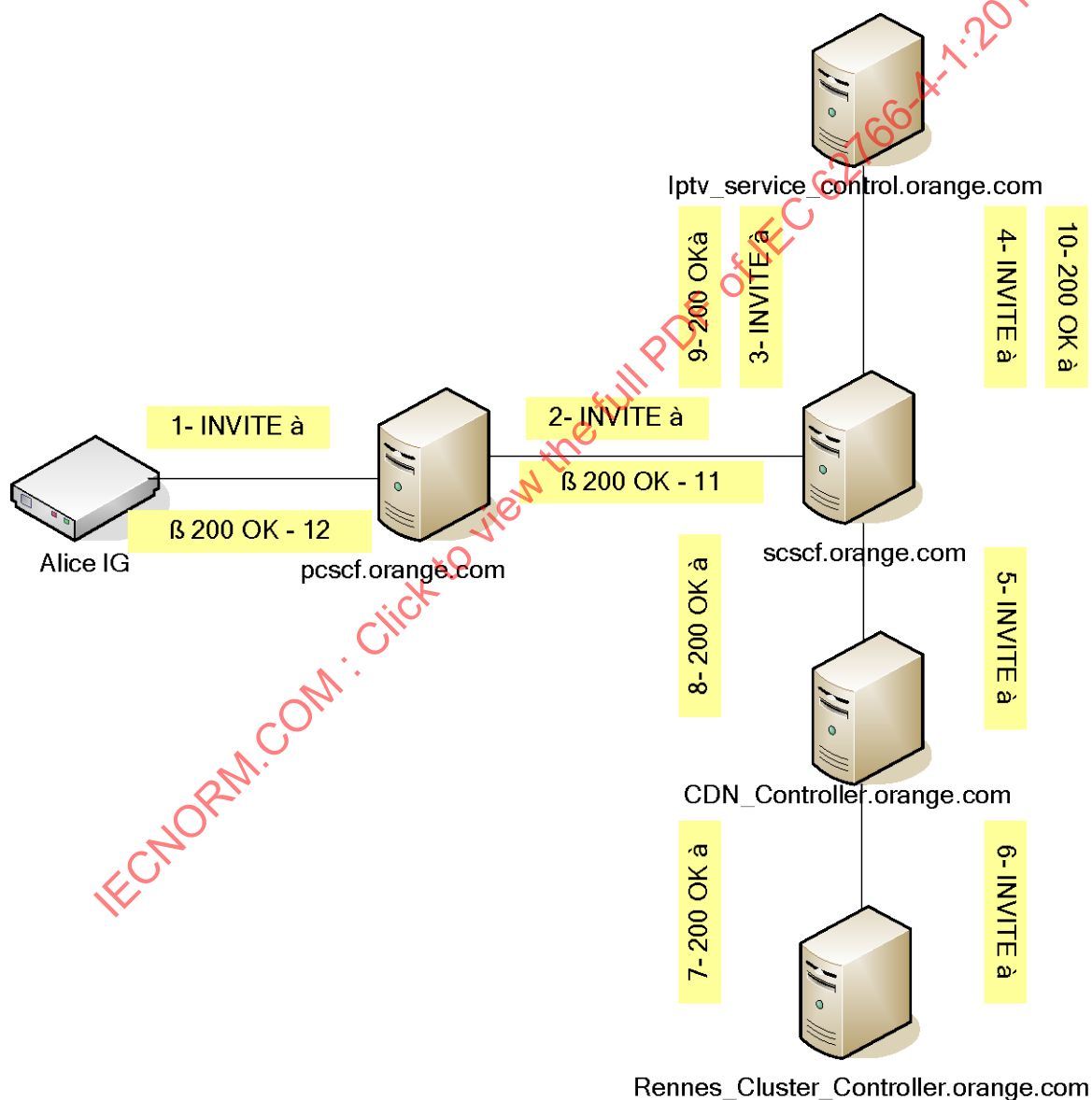


Figure A.1 – COD session setup sequence

IEC

NOTE The IG had received in the response to the REGISTER the service route e.g. <sip:pscsf.orange.com:5060;lr;comp=sigcomp> sip:scscf.orange.com:5060;lr; comp=sigcomp.

The following request_URI example is for the HD version of the movie “Twister”. In the BCG, Twister is signalled with the CRID: “CRID://warnerbros.com/Twister” and the HD instance is signalled with the IMI: “imi:HD”.

NOTE One or more of the characters “.”, “/” and “#” in the example below may need to be escaped as %2E (“.”), %2F (“/”) and %23 (“#”) depending on the restrictions imposed by the SIP request URI.

In the call flows below, unchanged information (after the = sign) in any step are left blank.

Step 1: Alice IG to P-CSCF:

```
INVITE sip: OIPF_IPTV_COD_SERVICE_warnerbros.com/Twister#HD@orange.com SIP/2.0
//CRID before @
Via: SIP/2.0/UDP 172.102.12.5:5061 //where to send the response
Max-Forwards: 70
Route: <sip:pcscf.orange.com:5060;lr;comp=sigcomp>,
      <sip:scscf.orange.com:5060;lr;comp=sigcomp>
From: AliceIG <sip:aliceIG@orange.com>;tag=1928301774 // tag for integrity
verification
To: sip: OIPF_IPTV_COD_SERVICE_warnerbros.com/Twister#HD@orange.com //same as
request URI
CSeq: 314159 INVITE
Contact: <sip: 172.102.12.5:5061;transport=UDP>
Content-Type: application/sdp
Content-Length: (...)

v=0
o=AliceIG 2890844527 2890844527 IN IP4 172.102.12.5
s=Streaming session
i=A Streaming session declared within the session description protocol
t=0
m=application 9 TCP iptv_rtsp // media line for RTSP control protocol
c=IN IP4 172.102.12.5
a=connection:new
a=setup:active

m=video 6666 RTP/AVP 33 // video
c=IN IP4 172.102.12.5
b= AS:4000
a=rcvonly
```

Step 2: P-CSCF to S-CSCF in ASM

```
INVITE
Via: SIP/2.0/UDP pcscf.orange.com:5060, SIP/2.0/UDP 172.102.12.5:5061
Max-Forwards: 69
Route: <sip:scscf.orange.com:5060;lr>
Record-Route: <sip:pcscf.orange.com:5060;lr;comp=sigcomp>
From:
To:
CSeq:
Contact:
Content-Type:
Content-Length: (...)

v=
o=
s=
i=
t=
m=
c=
a=
a=

m=
c=
b=
```

a=

Step 3: S-CSCF to IPTV Control

```

INVITE
Via: SIP/2.0/UDP scscf.orange.com:5060, SIP/2.0/UDP pcscf.orange.com:5060,
    SIP/2.0/UDP 172.102.12.5:5061,
Max-Forwards: 68
Record-Route: <sip:scscf.orange.com:5060;lr,comp=sigcomp>
    <sip:pcscf.orange.com:5060;lr;comp=sigcomp>
From:
To:
CSeq:
Contact:
Content-Type:
Content-Length: (..)
v=
o=
s=
i=
t=
m=
c=
a=
a=

m=
c=
b=
a=

```

Steps 4-5: IPTV Control to CDN controller via S-CSCF

```

INVITE sip:CDN_Controller@orange.com SIP/2.0

Max-Forwards: 66
Route: <sip:CDN_Controller@orange.com:5060;lr>
Record-Route: <sip:scscf.orange.com:5060;lr;comp=sigcomp>,
    <sip:IPTV_SCSCF@orange.com:5060;lr;comp=sigcomp>,
    <sip:scscf.orange.com:5060;lr;comp=sigcomp>,
    <sip:pcscf@orange.com:5060;lr;comp=sigcomp>
From:
To:
CSeq:
Contact:
Content-Type:
Content-Length: (..)
v=
o=
s=
i=
t=
m=
c=
a=
a=

m=
c=
b=
a=

```

Step 6: CDN controller to Rennes cluster controller

```
INVITE sip:Rennes_Cluster_Controller@orange.com SIP/2.0

Record-Route: <sip:CDN_Controller.orange.com:5060;lr,comp=sigcomp>,
               <sip:scscf.orange.com:5060;lr,comp=sigcomp>,
               <sip:IPTV_SCSCF.orange.com:5060;lr,comp=sigcomp>,
               <sip:scscf.orange.com:5060;lr,comp=sigcomp>,
               <sip:pcscf.orange.com:5060;lr,comp=sigcomp>
Max-Forwards: 65
Route: <sip:Rennes_Cluster_Controller.orange.com;lr>

From:
To:
CSeq:
Contact:
Content-Type:
Content-Length: (..)
v=
o=
s=
i=
t=
m=
c=
a=
a=
m=
c=
b=
a=
```

Step 7: Cluster Controller Reply to the CDN controller

```
SIP/2.0 200 OK

Via: SIP/2.0/UDP CDN_Controller.orange.com
Record-Route: <Rennes_Cluster_Controller.orange.com;lr,comp=sigcomp>

From:
To:
CSeq:
Contact:
Content-Type:
Content-Length: (..)

v=0
o= Rennes_Cluster_Controller IN IP4 Rennes_Cluster_Controller.orange.com
s=
i=
c=IN IP4 Rennes_Cluster_Controller.orange.com
t=0

m=application 999 TCP iptv_rtsp // media line for RTSP control protocol chosen
by CC
a=connection:new
a=setup:passive
a=fmtp:iptv_rtsp h-uri=rtsp://Rennes_Cluster_Controller.orange.com/Twister;
    h-session = 940211290776250

m=video 7777 RTP/AVP 33 // server video port
a=sendonly
```

Steps 8-12: 200 OK sent back to Alice IG using the same route.

A.2 Communication services message examples – HNI-IGI message mapping to SIP

A.2.1 General

The sample mappings provided in this subclause are focused on chat and Presence. They are not exhaustive.

The main use cases described in part 1 are covered:

- Presence
- Publication
- Subscription
- Notification
- chat
- Init
- Outgoing message (standard)
- Outgoing message (isComposing)
- incoming message
- Teardown

As an illustration, a typical IMS presence document is also presented in A.2.4.

IECNORM.COM : Click to view the full PDF of IEC 62766-4-1:2017

A.2.2 Presence

A.2.2.1 Initial publication

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP X-OITF-request-Line: PUBLISH sip:david@oiptv.org SIP/2.0 Host: 192.168.1.1 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Expires: 3600 X-OITF-Event: presence X-OITF-Call-Id: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-CSeq: 1 PUBLISH X-OITF-Content-Type: application/pidf+xml X-OITF-Content-length: (...) Content-Type: application/pidf+xml Content-Length: (...) <?xml version='1.0' encoding='UTF-8' ?> <presence xmlns='urn:ietf:params:xml:ns:pidf' entity='sip:david@oiptv.org'> ... </presence>	PUBLISH sip:david@oiptv.org SIP/2.0 Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=48240713 To: sip:david@oiptv.org CSeq: 1 PUBLISH Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Max-forwards: 10 Expires: 3600 Event: presence Content-Type: application/pidf+xml Content-Length: (...) <?xml version='1.0' encoding='UTF-8' ?> <presence xmlns='urn:ietf:params:xml:ns:pidf' entity='sip:david@oiptv.org'> ... </presence>
HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF- CSeq: 1 PUBLISH X-OITF- SIP-ETag: 1514024804 X-OITF- Expires: 6002 X-OITF-Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=48240713 To: sip:david@oiptv.org;tag=12ba5d-287-55-1366522802 CSeq: 1 PUBLISH Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Expires: 6002 SIP-ETag: 1514024804 Content-Length: 0

NOTE 1 Specifying both a 'From' and a 'To' allows an identity to publish on behalf of another identity.

NOTE 2 Here the server has requested a shorter expiration time that will be handled internally by the IG.

A.2.2.2 Updated publication

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP X-OITF-request-Line: PUBLISH sip:david@oiptv.org SIP/2.0 Host: 192.168.1.1 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Expires: 3600 X-OITF-Event: presence X-OITF- CSeq: 2 PUBLISH X-OITF- Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF- SIP-if-match: 15140248043 X-OITF- Content-Type: application/pidf+xml X-OITF- Content-length: (...) Content-Type: application/pidf+xml Content-Length: (...) <?xml version='1.0' encoding='UTF-8' ?> <presence xmlns='urn:ietf:params:xml:ns:pidf' entity='sip:david@oiptv.org'> ... </presence>	PUBLISH sip:david@oiptv.org SIP/2.0 Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=48240713 To: sip:david@oiptv.org CSeq: 2 PUBLISH Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Max-forwards: 10 SIP-if-match: 15140248043 Expires: 3600 Event: presence Content-Type: application/pidf+xml Content-Length: (...) <?xml version='1.0' encoding='UTF-8' ?> <presence xmlns='urn:ietf:params:xml:ns:pidf' entity='sip:david@oiptv.org'> ... </presence>
HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF- CSeq: 2 PUBLISH X-OITF- SIP-ETag: 7816034523 X-OITF- Expires: 600 X-OITF- Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=48240713 To: sip:david@oiptv.org;tag=12ba5d-287-55-1366522802 CSeq: 2 PUBLISH Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Expires: 600 SIP-ETag: 7816034523 Content-Length: 0

NOTE 3 SIP-IF-Match as retrieved from the previous publication acknowledgment.

A.2.2.3 End of publication

HNI-IGI Interface	SIP equivalent
POST IG_URI/SIP Host: 192.168.1.1 X-IOTF-request-Line: PUBLISH sip:david@oiptv.org SIP/2.0 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Expires: 0 X-OITF-Event: presence X-OITF-CSeq: 3 PUBLISH X-OITF- Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF- SIP-if-match: 78160345234 X-OITF- Content-Type: application/pdf+xml X-OITF- Content-length: 0 Content-Type: application/pdf+xml Content-Length: 0	PUBLISH sip:david@oiptv.org SIP/2.0 Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=48240713 To: sip:david@oiptv.org CSeq: 3 PUBLISH Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Max-forwards: 10 SIP-if-match: 78160345234 Expires: 0 Event: presence Content-Type: application/pdf+xml Content-Length: 0

HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-CSeq: 3 PUBLISH X-OITF-Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=48240713 To: sip:david@oiptv.org;tag=12ba5d-287-55-1366522802 CSeq: 3 PUBLISH Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Content-Length: 0
---	--

NOTE 4 Sip-if-match as retrieved from the previous publication acknowledgment.

A.2.2.4 Initial subscription

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP X-OITF-request-Line: SUBSCRIBE sip:fouz@oiptv.org SIP/2.0 Host: 192.168.1.1 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:fouz@oiptv.org X-OITF-Expires: 3600 X-OITF-Event: presence X-OITF-Accept: application/pdf+xml X-OITF- CSeq: 4 SUBSCRIBE X-OITF- Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF- Content-length: 0 Content-length: 0	SUBSCRIBE sip:fouz@oiptv.org SIP/2.0 Via: SIP/2.0/UDP 10.193.106.81:5060;branch=z9hG4bK1AD46E5D1E1 From: sip:david@oiptv.org;tag=2764425547 To: sip:fouz@oiptv.org CSeq: 4 SUBSCRIBE Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Contact: <sip:david@10.193.106.81:5060;transport=UDP> Expires: 3600 Event: presence Accept: application/pdf+xml Content-length: 0

HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip: fouz@oiptv.org X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-CSeq: 4 SUBSCRIBE X-OITF-Expires: 6005 X-OITF-Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=2764425547 To: sip:fouz@oiptv.org;tag=12ba5d-287-55-1366522802 CSeq: 4 SUBSCRIBE Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Expires: 6005 Content-Length: 0
---	--

NOTE 5 Here the server has requested a shorter expiration time that will be handled internally by the IG.

A.2.2.5 Notification (individual)

HNI-IGI Interface	SIP equivalent
HTTP 200 OK X-OITF-response-Line: NOTIFY sip:david@10.193.106.72:5060;transport=UDP SIP/2.0 X-OITF-From: sip:fouz@oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-Event: presence X-OITF-CSeq: 1001 NOTIFY X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-Subscription-State: active; expires=600 X-OITF-Content-Type: application/pidf+xml X-OITF-Content-Length: (...) Content-Type: application/pidf+xml Content-Length: (...) <?xml version="1.0" encoding="UTF-8"?> <presence xmlns="urn:ietf:params:xml:ns:pidf" entity="sip:fouz@oiptv.org"> ... </presence>	NOTIFY sip:david@10.193.106.72:5060;transport=UDP SIP/2.0 Via: SIP/2.0/UDP 10.194.117.18:5060;branch=z9hG4bK0014c262ba5d-2 From: sip:fouz@oiptv.org;tag=10014c262ba5d To: sip:david@oiptv.org;tag=2764425547 CSeq: 1001 NOTIFY Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Subscription-State: active; expires=600 Contact: <sip:10.194.117.18:5060;transport=UDP> Event: presence Content-Type: application/pidf+xml Content-Length: (...) <?xml version="1.0" encoding="UTF-8"?> <presence xmlns="urn:ietf:params:xml:ns:pidf" entity="sip:fouz@oiptv.org"> ... </presence>

NOTE An acknowledgment from the IG to the network is required but not described here.

A.2.2.6 Subscription refresh

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP X-OITF-request-Line: SUBSCRIBE sip:fouz@oiptv.org SIP/2.0 Host: 192.168.1.1 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:fouz@oiptv.org X-OITF-Expires: 3600 X-OITF-Event: presence X-OITF-Accept: application/pidf+xml X-OITF-CSeq: 5 SUBSCRIBE X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-Contact: <sip:david@10.193.106.81:5060;transport=UDP> X-OITF-Content-length: 0 Content-length: 0	SUBSCRIBE sip:fouz@oiptv.org SIP/2.0 Via: SIP/2.0/UDP 10.193.106.81:5060;branch=z9hG4bK1AD46E5D1E1 From: sip:david@oiptv.org;tag=2764425547 To: sip:fouz@oiptv.org CSeq: 5 SUBSCRIBE Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Contact: <sip:david@10.193.106.81:5060;transport=UDP> Expires: 3600 Event: presence Accept: application/pidf+xml Content-length: 0
HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip: fouz@oiptv.org X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-CSeq: 5 SUBSCRIBE X-OITF-Expires: 6005 X-OITF-Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=2764425547 To: sip:fouz@oiptv.org;tag=12ba5d-287-55-1366522802 CSeq: 5 SUBSCRIBE Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Expires: 6006 Content-Length: 0

NOTE 1 From the OITF's point-of-view, both initial and refresh subscription requests are identical.

NOTE 2 Here the server has requested a shorter expiration time, which will be handled internally by the IG.

A.2.2.7 End of subscription

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP X-OITF-request-Line: SUBSCRIBE sip:fouz@oiptv.org SIP/2.0 Host: 192.168.1.1 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:fouz@oiptv.org X-OITF-Expires: 0 X-OITF-Event: presence X-OITF-Accept: application/pidf+xml X-OITF-CSeq: 6 SUBSCRIBE X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-Contact: <sip:david@10.193.106.81:5060;transport=UDP> X-OITF-Content-length: 0 Content-length: 0	SUBSCRIBE sip:fouz@oiptv.org SIP/2.0 Via: SIP/2.0/UDP 10.193.106.81:5060;branch=z9hG4bK1AD46E5D1E1 From: sip:david@oiptv.org;tag=2764425547 To: sip:fouz@oiptv.org CSeq: 6 SUBSCRIBE Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Contact: <sip:david@10.193.106.81:5060;transport=UDP> Expires: 0 Event: presence Accept: application/pidf+xml Content-length: 0

HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip: fouz@oiptv.org X-OITF-Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x X-OITF-CSeq: 5 SUBSCRIBE X-OITF-Expires: 6005 X-OITF-Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK Via: SIP/2.0/UDP 10.194.56.134:5060;branch=z9h4bK61C529E16989 From: sip:david@oiptv.org;tag=2764425547 To: sip:fouz@oiptv.org;tag=12ba5d-287-55-1366522802 CSeq: 6 SUBSCRIBE Call-ID: 78A0g080Ca3502i6414m360Bt38A6b2E4Fx61C8x Content-Length: 0
--	---

A.2.3 Chat**A.2.3.1 Chat session setup**

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP Host: 192.168.1.1 X-OITF-request-Line: INVITE sip:sports.room@chat.oiptv.org SIP/2.0 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:sports.room@chat.oiptv.org X-OITF-Accept: message/cpim X-OITF-Call-ID: 3413an89KU X-OITF-Content-Type: application/sdp X-OITF-Content-length: (...) Content-length: 0	INVITE sip:sports.room@chat.oiptv.org SIP/2.0 To: sip:sports.room@chat.oiptv.org From: sip:david@oiptv.org;tag=786 Call-ID: 3413an89KU Content-Type: application/sdp Content-length: (...) c=IN IP4 10.194.52.13 m=message 7654 TCP/MSRP * a=accept-types:message/cpim a=path:msrp://10.194.52.13:7654/jshA7weztas;tcp

HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oipvtv.org X-OITF-To: sip:sports.room@chat.oipvtv.org X-OITF-Call-ID: 3413an89KU X-OITF-Content-Type: application/sdp X-OITF-Accept: message/cpim X-OITF-Content-length: (...) Content-Length: 0	SIP/2.0 200 OK To: sip:sports.room@chat.oipvtv.org;tag=087js From: sip:david@oipvtv.org;tag=786 Call-ID: 3413an89KU Content-Type: application/sdp Content-length: (...) c=IN IP4 chat.oipvtv.org m=message 12763 TCP/MSRP * a=accept-types:message/cpim a=path:msrp://chat.oipvtv.org:12763/kjhd37s2s20w2a;tcp
--	---

NOTE A final acknowledgment from the IG to the network is required, but not described here

A.2.3.2 Chat outgoing message (standard)

HNI-IGI interface	MSRP equivalent
POST IG_URI/AUX X-HNI-IGI-request: MSRP SEND MESSAGE X-HNI-IGI-message-ID: X-HNI-IGI-From: sip:david@oipvtv.org X-HNI-IGI-To: sip:sports.room@chat.oipvtv.org Who else thinks this late penalty was a disgrace?	MSRP a786hjs2 SEND To-Path: msrp://chat.oipvtv.org:12763/kjhd37s2s20w2a;tcp8 From-Path: msrp://10.194.52.13:7654/jshA7weztas;tcp8 message-ID: 87652491 Byte-Range: (...) Content-Type: message/cpim To: sip:sports.room@chat.oipvtv.org From: David <sip:david@oipvtv.org> DateTime: 2008-06-15T15:02:31-03:00 Content-Type: text/plain Who else thinks this late penalty was a disgrace? -----a786hjs2\$
HTTP/1.1 200 OK X-HNI-IGI-response-Line: MSRP 200 OK X-HNI-IGI-message-ID: 87652491 X-HNI-IGI-From: sip:david@oipvtv.org X-HNI-IGI-To: sip:sports.room@chat.oipvtv.org Content-Length: 0	MSRP a786hjs2 200 OK To-Path: msrp://chat.oipvtv.org:12763/kjhd37s2s20w2a;tcp From-Path: msrp://10.194.52.13:7654/jshA7weztas;tcp -----a786hjs2\$

The IG is responsible for mapping the caller and callee URIs to the actual MSRP paths exchanged during the chat setup.

A.2.3.3 Chat outgoing message (isComposing)

HNI-IGI interface	MSRP equivalent
POST IG_URI/AUX X-HNI-IGI-request: MSRP SEND ACTIVITY X-OITF-message-ID: 87653492 X-HNI-IGI-From: sip:david@oipvtv.org X-HNI-IGI-To: sip:sports.room@chat.oipvtv.org <?xml version="1.0" encoding="UTF-8"?> <isComposing xmlns="urn:ietf:params:xml:ns:im-iscomposing" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:ietf:params:xml:ns:im-composing iscomposing.xsd"> <state>active</state> <contenttype>text/plain</contenttype> <refresh>90</refresh> </isComposing>	MSRP a786hjs2 SEND To-Path: msrp://chat.oipvtv.org:12763/kjhd37s2s20w2a;tcp9 From-Path: msrp://10.194.52.13:7654/jshA7weztas;tcp9 message-ID: 87652492 Byte-Range: (...) Content-Type: message/cpim Content-Length: (...) To: sip:sports.room@chat.oipvtv.org From: David <sip:david@oipvtv.org> DateTime: 2008-06-15T15:02:31-03:00 Content-Type: application/im-iscomposing+xml <?xml version="1.0" encoding="UTF-8"?> <isComposing xmlns="urn:ietf:params:xml:ns:im-iscomposing" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:ietf:params:xml:ns:im-composing iscomposing.xsd"> <state>active</state> <contenttype>text/plain</contenttype> <refresh>90</refresh> </isComposing> -----a786hjs2\$
HTTP/1.1 200 OK X-HNI-IGI-response-Line: MSRP 200 OK X-HNI-IGI-message-ID: 87652491 X-HNI-IGI-From: sip:david@oipvtv.org X-HNI-IGI-To: sip:sports.room@chat.oipvtv.org Content-Length: 0	MSRP a786hjs2 200 OK To-Path: msrp://chat.oipvtv.org:12763/kjhd37s2s20w2a;tcp From-Path: msrp://10.194.52.13:7654/jshA7weztas;tcp -----a786hjs2\$

The IG is responsible for mapping the caller and callee URIs to the actual MSRP paths exchanged during the chat setup.

A.2.3.4 Chat incoming message

HNI-IGI interface	MSRP equivalent
HTTP/1.1 200 OK X-HNI-IGI-request: MSRP RECEIVED MESSAGE X-OITF-From: sip:sports.room@chat.oiptv.org X-OITF-To: sip:david@oiptv.org X-OITF-message-ID: 56712483 I don't care: we won anyway!	MSRP a786hjs2 SEND To-Path: msrp://10.194.52.13:7654/jshA7weztas;tcp10 From-Path: msrp://chat.oiptv.org:12763/kjhd37s2s20w2a;tcp10 message-ID: 56712483 Byte-Range: (...) Content-Type: message/cpim Content-Length: (...) To: sip:sports.room@chat.oiptv.org From: Fouz <sip:fouz@oiptv.org> DateTime: 2008-06-15T15:02:31-03:00 Content-Type: text/plain I don't care: we won anyway! -----a786hjs2\$

NOTE An acknowledgment from the IG to the network is required, but not described here.

The IG shall be responsible for mapping the MSRP paths exchanged during the chat setup to the actual caller and callee URIs.

A.2.3.5 Chat session teardown

HNI-IGI interface	SIP equivalent
POST IG_URI/SIP X-OITF-request-Line: BYE sip:sports.room@chat.oiptv.org SIP/2.0 Host: 192.168.1.1 X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:sports.room@chat.oiptv.org X-OITF-Call-ID: 3413an89KU11 X-OITF-CSeq: 231 BYE X-OITF-Content-Length: 0 Content-length: 0	BYE sip:sports.room@chat.oiptv.org SIP/2.0 Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10 Max-Forwards: 70 To: sip:sports.room@chat.oiptv.org;tag=087js From: sip:david@oiptv.org;tag=786 Call-ID: 3413an89KU11 CSeq: 231 BYE Content-Length: 0
HTTP/1.1 200 OK X-OITF-response-Line: SIP/2.0 200 OK X-OITF-From: sip:david@oiptv.org X-OITF-To: sip:sports.room@chat.oiptv.org X-OITF-Call-ID: 3413an89KU X-OITF-CSeq: 231 BYE X-OITF-Content-Length: 0 Content-Length: 0	SIP/2.0 200 OK To: sip:sports.room@chat.oiptv.org;tag=087js From: sip:david@oiptv.org;tag=786 Call-ID: 3413an89KU CSeq: 231 BYE Content-length: 0

A.2.4 Presence document

A.2.4.1 Presence schema

See Annex I for the Presence XML schema.

A.2.4.2 Presence schema examples

A.2.4.2.1 General

Examples of how the presence information semantics are described in a typical presence Information XML schema are shown below.

A.2.4.2.2 Example of presence for broadcast TV service

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:iptv="urn:oipf:service:oitfpresence:2011"
  xmlns:oma="urn:oma:xml:prs:pidf:oma-pres"
  xmlns:pdm="urn:ietf:params:xml:ns:pidf:data-model"
  xmlns="urn:ietf:params:xml:ns:pidf" entity="sip:someone@example.com">
  <tuple id="abcde">
    <status>
      <basic>open</basic>
    </status>

    <oma:service-description>
      <oma:service-id>Broadcast-TV</oma:service-id>
      <oma:version>1.0</oma:version>
      <oma:description>Broadcast TV service</oma:description>
    </oma:service-description>
    <iptv:BroadcastTVservice>
      <iptv:BroadcastTV Technology="DVB-T">
        <iptv:currentChannel>BBC</iptv:currentChannel>
        <iptv:currentProgram>News</iptv:currentProgram>
        <iptv:serviceID>BBC_ID</iptv:serviceID>
      </iptv:BroadcastTV>
    </iptv:BroadcastTVservice>

    <timestamp>2008-07-08T12:34:21Z</timestamp>
  </tuple>

  <pdm:device id="aa111">
    <pdm:deviceID>
      urn:uuid:11162e19-5fbf-43fc-a2fd-d23002787599
    </pdm:deviceID>
    <pdm:timestamp>2008-07-08T12:34:21Z</pdm:timestamp>
  </pdm:device>
</presence>
```

Annex B (informative)

User profile description

B.1 IPTV subscription profile

B.1.1 General

IPTV subscription profile encompasses relevant information required to operate an IPTV service. This includes user settings regarding:

- global settings (language preference, user action recordable);
- broadcast settings, with list of subscribed scheduled content service packages.

Broadcast service refers to scheduled content services. Accordingly, broadcast settings refer to the scheduled content settings.

A scheduled content service package is a set of elementary scheduled content TV services, along with a description. These scheduled content services have the same authorization and charging policy.

A scheduled content IPTV service is for instance a multicast IPTV channel, interactive channel, mosaic to which a user may subscribe.

The broadcast settings only provide a reference to service package and/or associated services to which a given IPTV user has subscribed, and is not meant to be a complete description of the service package and/or service. The complete service package and/or service description should be available in an associated IPTV service profile definition. If the list of elementary IPTV services associated with a given service package are not explicitly listed in the IPTV subscription profile, then it implies that the IPTV user has implicitly subscribed to all of the IPTV services within that service package.

- Content on demand settings (parental control level).
- PVR settings (PVR preferences network/local, PVR user restrictions, PVR storage limit).
- User equipment information (OITF) which uniquely identifies the user's OITF, classifies it as a device type (OITF-STV, OITF-TV) and provides relevant device capabilities. An IPTV user may be associated with one or more OITF(s) and every OITF is uniquely identified with a Unique Identifier (tUEID). The OITF capabilities associated with an IPTV user profile may be used for customization of IPTV service selection data that is provided by the IPTV service discovery to the IPTV user (based on capabilities of the OITF with which the IPTV user is currently associated). For instance, an IPTV user on a SD-only device would not be provided with information related to HD services. The OITF settings is not intended to cover all information related to the OITF and currently holds only the OITF capabilities attribute since this information may be used by the IPTV service discovery for personalized service selection.

NOTE Detailed information about the OITF may be located elsewhere and can be referenced by the IPTV Subscription profile using the tUEID element.

B.1.2 XML Schema for the IPTV subscription profile

XML schema for the IPTV profile, based on ETSI TS 183 063, Annex C:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="org:oipf:iptv:IPTVprofile:2011"
  xmlns:tns="org:oipf:iptv:IPTVprofile:2011"
  xmlns:ueprofile="org:oipf:iptv:UEprofile:2010"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:import namespace="org:oipf:iptv:UEprofile:2010"
    schemalocation="iptv-UEprofile.xsd" />
  <xs:element name="IPTVprofile">
    <xs:annotation>
      <xs:documentation>
        XML Schema for representing the IPTV profile object
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="UEprofile" type="ueprofile:tUEprofile"
minOccurs="0" />
        <xs:element name="GlobalSettings" type="tns:tGlobalSettings" />
        <xs:element name="BCprofile" type="tns:tBCprofile" minOccurs="0"
/>
        <xs:element name="CoDprofile" type="tns:tCoDprofile"
minOccurs="0" />
        <xs:element name="PVRprofile" type="tns:tPVRprofile"
minOccurs="0" />
        <xs:element name="extension" type="tns:textension" minOccurs="0"
/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="profileId" type="xs:ID" />
      <xs:anyAttribute/>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="tBCprofile">
    <xs:sequence>
      <xs:element name="BCservicePackage" type="tns:tBCservicePackage"
maxOccurs="unbounded" />
      <xs:element name="extension" type="tns:textension" minOccurs="0"
/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tBCservicePackage">
    <xs:sequence>
      <xs:element name="BCPackageId" type="tns:tBCservicePackageID" />
      <xs:element name="Description"
type="tns:tBCservicePackageDescription"
minOccurs="0" />
      <xs:element name="BCservice" type="tns:tBCservice" minOccurs="0"
maxOccurs="unbounded" />
      <xs:element name="extension" type="tns:textension" minOccurs="0"
/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

```