OIPF

Release 2 Specification

Volume 4 – Protocols

[V2.3] – [2014-01-24]

Open IPTV Forum

Open IPTV Forum

Postal address

Open IPTV Forum support office address

650 Route des Lucioles – Sophia Antipolis
Valbonne – FRANCE
Tel.: +33 4 92 94 43 83
Fax: +33 4 92 38 52 90

Internet

http://www.oipf.tv

Disclaimer

The Open IPTV Forum accepts no liability whatsoever for any use of this document.

Copyright Notification

Contents

Tables

Figures

Foreword

This Technical Specification (TS) has been produced by the Open IPTV Forum.

This specification provides multiple options for some features. The Open IPTV Forum Profiles specification complements the Release 2 specifications by defining the Open IPTV Forum implementation and deployment profiles.

This document is Volume 4 in the 10 Volume set of specifications that define the Open IPTV Forum Release 2 Solution. Other Volumes in the set are:

- Volume 1 – Overview
- Volume 2 – Media Formats
- Volume 2a – HTTP Adaptive Streaming
- Volume 3 – Content Metadata
- Volume 4a – Examples of IPTV Protocol Sequences
- Volume 5 – Declarative Application Environment
- Volume 5a – Web Standards TV Profile
- Volume 6 – Procedural Application Environment
- Volume 7 – Authentication, Content Protection and Service Protection

Introduction

This document specifies the protocols over the following reference point interfaces defined in the Open IPTV Forum Release 2 Architecture specification [OIPF_ARCH2].

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

The requirements for these interfaces are derived from the following sources:

- Open IPTV Forum Service and Platform Requirement for Release 2 [OIPF_REQS2]
- Open IPTV Forum Functional Architecture for Release 2 [OIPF_ARCH2]
- Other Open IPTV Forum specifications [OIPF_DAE2], [OIPF_CSP2], [OIPF_META2] and [OIPF_MEDIA2]

1 References

1.1 Normative References

[TS124503]	ETSI, TS 124 503, “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” [3GPP TS 24.229 (Release 7), modified] (3GPP TS 24.503 Release 8)
[UMTS-SH]	ETSI, TS 129 329, “Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Sh interface based on the Diameter protocol; Protocol details” (3GPP TS 29.329 version 7.4.0 Release 7)
[CHNG]	ETSI, ES 282 010, “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Charging”
[DIAM]	ETSI, TS 183 033, “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” [3GPP TS 29.228 V6.8.0 and 3GPP TS 29.229 V6.6.0, modified]
[AFSPDF]	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”
[RACS-RE]	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”
[NASS-E4]	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”
[HTTP]	IETF, RFC 2616, “Hypertext Transfer Protocol -- HTTP/1.1”
[SIP]	IETF, RFC 3261, “SIP: Session Initiation Protocol”
[BCG]	ETSI, TS 102 539, “Digital Video Broadcasting (DVB);Carriage of Broadband Content Guide (BCG) information over Internet Protocol (IP)”
[TR069]	Broadband Forum, TR-069 Amendment 2, “CPE WAN Management Protocol v1.1”
[SIP-PRES]	IETF, RFC 3856, “A Presence Event Package for the Session Initiation Protocol (SIP)”
[TS183019]	ETSI, TS 183 019, “Network Attachment: User-Network protocol Interface Definitions”
[SIP-EVNT]	IETF, RFC 3265, “Session Initiation Protocol (SIP)-Specific Event Notification”
[TS183063]	ETSI, TS 183 063, “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN);IMS-based IPTV stage 3 specification”, Release 3
[TS102034]	ETSI, TS 102 034 V1.5.1, “Digital Video Broadcasting (DVB);Transport of MPEG-2 TS Based DVB Services over IP Based Networks”
[FCC]	DVB BlueBook A152 (08/10), “Server-Based Fast Channel Change in DVB-IPTV Systems”
[PORTMAP]	IETF, RFC 6284, “Port Mapping Between Unicast and Multicast RTP Sessions”
[XCAP]	IETF, RFC 4825, “The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)”
[RFC3840]	IETF, RFC 3840, “Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)”
[RFC3455]	IETF, RFC 3455, “Private Header (P-Header) Extensions to the Session Initiation. Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)”

[SIP-IM]	IETF, RFC 3428, “Session Initiation Protocol (SIP) Extension for Instant Messaging”
[RTP]	IETF, RFC 3550, “RTP: A Transport Protocol for Real-Time Applications”
[TVA]	ETSI, TS 102 822-4, “Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems (“TV-Anytime Phase 1”); Part 4:Content referencing”
[TVA-MD]	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”
[SDP-TCP]	IETF, RFC 4145, “TCP-Based Media Transport in the Session Description Protocol (SDP)”
[SIP-REG]	IETF, RFC 3680, “A Session Initiation Protocol (SIP) Event Package for Registrations”
[RFC3841]	IETF, RFC 3841, “Caller Preferences for the Session Initiation Protocol (SIP)”
[SMPL-IM]	OMA, OMA-TS-SIMPLE_IM-V1_0-20100322-C, “Instant Messaging using SIMPLE”
[SMPL-PRES]	OMA, OMA-ERP-Presence_SIMPLE-V1_1-20080627-A, “Presence SIMPLE Specification”
[RTSP]	IETF, RFC 2326, “Real Time Streaming Protocol (RTSP)”
[RTSP2-AN]	IETF, draft-stiemerling-rtsp-announce-01, “RTSP 2.0 Asynchronous Notification”
[IGMP3]	IETF, RFC 3376, “Internet Group Management Protocol, Version 3”
[DLNA]	DLNA, IEC, 62481-2, Digital living network alliance (DLNA) home networked device interoperability guidelines – Part 2: Media Formats, ed1.0 (2007-08).
[GAA]	3GPP, TS 33.220, “Generic Authentication Architecture (GAA); Generic bootstrapping architecture”
[UB-UA]	3GPP, TS 24.109, “Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details”
[ADDR]	IETF, RFC 1918, “Address Allocation for Private Internets”
[3G-SEC]	3GPP, TS 33.203, “3G security; Access security for IP-based services”
[XCAP-EVT]	IETF, RFC 5875, “An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Diff Event Package”
[XCAP-DFF]	IETF, RFC 5874, “An Extensible Markup Language (XML) Document Format for Indicating A Change in XML Configuration Access Protocol (XCAP) Resources”
[CEA-2014-A]	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.
[CLSLESS]	IETF, RFC 3442, “The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4”
[DHCP-OPT]	IETF, RFC 2132, “DHCP Options and BOOTP Vendor Extensions”
[DHCP-VND]	IETF, RFC 3925, “Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)”
[DOM-NAME]	IETF, RFC 1035, “Domain Names - Implementation And Specification”
[SDP-RTCP]	IETF, RFC 3556, “Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth”
[SES-TIMR]	IETF, RFC 4028, “Session Timers in the Session Initiation Protocol (SIP)”
[UPNP]	UPnP Forum, “UPnP Device Architecture”
[RTCP-XR]	IETF, RFC 3611, “RTP Control Protocol Extended Reports (RTCP XR)”
[PSS]	3GPP, TS 26.234v750, “Transparent end to end packet switched streaming service (PSS) – Protocols and Codecs”
[FEC]	IETF, RFC 4756, “Forward Error Correction Grouping Semantics in Session Description Protocol”

[SIP-CFG]	IETF, RFC 6080, “A Framework for Session Initiation Protocol User Agent Profile Delivery”
[RFC3994]	IETF, RFC 3994, “Indication of Message Composition for Instant Messaging”
[RFC3551]	IETF, RFC 3551, “RTP Profile for Audio and Video Conferences with Minimal Control”
[TR135]	Broadband Forum, TR-135, “Data Model for a TR-069 Enabled STB”
[TR106]	Broadband Forum, TR-106, “Data Model Template for TR-069 Enabled Devices”
[TR098]	Broadband Forum, TR-098, “Internet Gateway Device Data Model for TR-069”
[TR104]	Broadband Forum, TR-104, “DSLHome™ Provisioning Parameters for VoIP CPE”
[RFC3926]	IETF, RFC 3926, “FLUTE - File Delivery over Unidirectional Transport”
[SHA-1]	U.S. Department of Commerce/National Institute of Standards and Technology, FIPS PUB 180-1, “Secure Hash Standard”
[RFC4961]	IETF, RFC 4961, “Symmetric RTP/RTP Control Protocol (RTCP)”
[RFC4787]	IETF, RFC 4787, “Network Address Translation (NAT) Behavioural Requirements for Unicast UDP”
[ES282003]	ETSI, ES 282 003, “Resource and Admission Control Sub-system (RACS) - Functional architecture”
[SDP]	IETF, RFC 4566, “SDP: Session Description Protocol”
[RFC3986]	IETF, RFC 3986, “Uniform Resource Identifier (URI): Generic Syntax”
[RFC5621]	IETF, RFC 5621, “Message Body Handling in the Session Initiation Protocol (SIP)”
[RFC5626]	IETF, RFC 5626, “Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)”
[RFC4122]	IETF, RFC 4122, “A Universally Unique Identifier (UUID) Namespace”
[INFO-PKG]	IETF, RFC 6086, “Session Initiation Protocol (SIP) INFO Method and Package Framework”
[PSS-MBMS]	3GPP, TS 26.237, “IP Multimedia Subsystem (IMS) based Packet Switch Streaming (PSS) and Multimedia Broadcast/Multicast Service (MBMS) User Service; Protocols (Release 8)”
[ParlayXSMS]	3GPP, TS 29.199-4, “Open Service Access (OSA); Parlay X web services; Part 4: Short messaging”
[RFC2837]	IETF, RFC 2387, “The MIME Multipart/Related Content-type”
[RFC3515]	IETF, RFC 3515, “The Session Initiation Protocol (SIP) REFER Method
[RFC3420]	IETF, RFC 3420, “Internet Media Type message/sipfrag”
[RFC3891]	IETF, RFC 3891, “The Session Initiation Protocol (SIP) “Replaces” Header”
[RFC5627]	IETF, RFC 5627, “Obtaining and Using Globally Routable User Agent URIs (GRUU) in the Session Initiation Protocol (SIP)”
[UPNP-MR]	UPnP Forum, “MediaRenderer:1 Device Template Version 1.01”
[UPNP-MS]	UPnP Forum, “MediaServer:1 Device Template Version 1.01”
[RFC4975]	IETF, RFC 4975, “The Message Session Relay Protocol (MSRP)”
[RFC3588]	IETF, RFC 3588, “Diameter Base Protocol”
[DIAMCHG]	3GPP, TS 32.299, “Telecommunication management; Charging management; Diameter charging applications (Release 8)”
[SRVCONT]	3GPP, TS 24.237, “IP Multimedia (IM) Core Network (CN) subsystem IP Multimedia Subsystem (IMS) service continuity; Stage 3”
[HTTPAUTH]	IETF, RFC 2617, “HTTP Authentication: Basic and Digest Access Authentication”

[OFRANSR]	IETF, RFC 3264, “An Offer/Answer Model with the Session Description Protocol (SDP)”
[RFC3551]	IETF, RFC 3551, “RTP Profile for Audio and Video Conferences with Minimal Control”, July 2003
[RFC3984]	IETF, RFC 3984, “RTP Payload Format for H.264 Video”, February 2005
[RFC3016]	IETF, RFC 3016, “RTP Payload Format for MPEG-4 Audio/Visual Streams”, 2000
[RFC4629]	IETF, RFC 4629, “RTP Payload Format for ITU-T Rec. H.263 Video”, 2007
[RFC4867]	IETF, RFC 4867, “RTP Payload Format and File Storage Format for the AMR and AMR-WB Audio Codecs”, 2007
[RFC4749]	IETF, RFC 4749, “RTP Payload Format for the G.729.1 Audio Codec”, October 2006
[RFC5404]	IETF, RFC 5404, “RTP Payload Format for G.719”, January 2009
[RFC3640]	IETF, RFC 3640, “RTP Payload Format for Transport of MPEG-4 Elementary Streams”, November 2003
[VIDEOSHARE]	GSMA, PRD IR.84 Video Share Interoperability Specification, 2.0.
[TS26114]	3GPP, TS 26.114, “IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction”.
[TS24173]	3GPP, TS 24.173, “IMS multimedia telephony communication service and supplementary services; Stage 3”.
[TS181005]	ETSI, TS 181 005, “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Service and Capability Requirements".
[RFC3890]	IETF, RFC 3890, “A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)”
[RFC4588]	IETF, RFC 4588, “RTP Retransmission Payload Format”
[RFC4605]	IETF, RFC 4605, “Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")”
[RFC4867]	IETF, RTF 4867, “RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs”
[RFC4627]	IETF, RFC 4627, “The application/json Media Type for JavaScript Object Notation (JSON)”
RFC3984]	IETF, RFC 3984, “RTP Payload Format for H.264 Video”
[RFC2045]	IETF, RFC 2045, “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”
[ES283002]	ETSI, ES 283 002, “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN);H.248 Profile for controlling Access and Residential Gateways”
[AAC]	ISO/IEC 14496-3:2009, “Information Technology – Coding of audio-visual objects – Part 3: Audio”.
[RFC3903]	IETF, RFC 3903, “Session Initiation Protocol (SIP) Extension for Event State Publication”
[ES283003]	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]
[MPEG2TS]	ISO/IEC 13818-1:2000/Amd.3:2004, “Generic coding of moving pictures and associated audio information: Systems”
[RAMS]	IETF, RFC 6285, “Unicast-Based Rapid Acquisition of Multicast RTP Sessions”
[DVB-SC]	ETSI, TS 103 197 V1.5.1, “Digital Video Broadcasting (DVB); Head-end implementation of DVB SimulCrypt”, March 2007.

1.2 Open IPTV Forum References

[OIPF_REQS2]	Open IPTV Forum, “Service and Platform Requirements”, V2.0, December 2008.
[OIPF_ARCH2]	Open IPTV Forum, “Functional Architecture - Version 2.3”, January 2014.
[OIPF_MEDIA2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 2 - Media Formats”, V2.3, January 2014.
[OIPF_META2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 3 - Content Metadata”, V2.3, January 2014.
[OIPF_PROT2_EX]	Open IPTV Forum, “Release 2 Solution Specification, Volume 4a - Examples of IPTV Protocol Sequences”, V2.3, January 2014.
[OIPF_DAE2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 5 - Declarative Application Environment”, V2.3, January 2014.
[OIPF_CSP2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 7 - Authentication, Content Protection and Service Protection”, V2.3, January 2014.

1.3 Informative References

[RFC2119]	IETF, RFC 2119, “Key words for use in RFCs to Indicate Requirement Levels”
[ABNF]	IETF, RFC 4234, “Augmented BNF for Syntax Specifications: ABNF”

2 Conventions and Terminology

2.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [RFC2119]. All sections and annexes, except “Introduction”, are normative, unless they are explicitly indicated to be informative.

OIPF usually abbreviates "Scheduled Content" as "SC". This specification additionally uses, for the same meaning and only in the field names, the abbreviation "BC".

2.2 Terminology

2.2.1 Definitions

Term	**Definition**
Native HNI-IGI function (often shortened to Native HNI-IGI)	The procedures for interactions on the HNI-IGI interface are provided as part of the OITF implementation - typically in native code.
Non-native HNI-IGI function (often shortened to Non-native HNI-IGI)	The procedures for interactions on the HNI-IGI interface are provided by a service provider in JavaScript as part of a DAE application.

2.2.2 Abbreviations

In addition to the abbreviations provided in Volume 1, the following abbreviations are used in this Volume.

Abbreviation	**Definition**
FCC	Fast Channel Change
GSMA	GSM Association
ISC	IMS Service Control
PCh	Personalised Channel
PPV	Pay Per View
RET	RETransmission Function
RFC	Request For Comments
XCAP	XML Configuration Access Protocol
XDM	XML Document Management

3 Release 2 Interfaces

3.1 Consumer Network to Provider Network Interfaces (UNI)

The functional entities, functions and reference points defined for the Consumer Network are described in section 5.3 of [OIPF_ARCH2].

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 [BCG].	HTTP, DVBSTP
UNIS-8	Authentication and session management relying on IMS.	IMS SIP
UNIS-9	Authentication for GBA Single-Sign on. See [OIPF_CSP2]	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 subject to standardization.	HTTP, FLUTE
UNIS-13	User Stream control for multicast of real time content and data.	IGMP
UNIS-14	Reference point used for authorization of serviceaccess. See [OIPF_CSP2]	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 [TR069] 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 [OIPF_CSP2]	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: 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]
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 RFC 3261 [SIP] and pertinent 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 a DAE applications and MUST 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 which MUST 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 MUST be implemented natively consist of user registration (sections 5.4.6.1 and 6.1.3.2.2) including service provider discovery (section 5.4.1.1), and GBA procedures (section 5.4.6.2) 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 [OIPF_CSP2] section 5.4.5 or the, more general, procedure, HTTP Digest Authentication using IMS Gateway in [OIPF_CSP2] section 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.

3.2 Provider Network Reference Points Description

The Functional Entities and Reference Points in the service provider network defined in section 5.2 of [OIPF_ARCH2].

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 [TS124503]
NPI-4	Reference point for routing of IPTV service related messages to the IPTV Control FE.	ISC interface [TS124503]
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 [UMTS-SH]
NPI-9	This reference point allows the IPTV Control Point to retrieve the subscriber's IMS-specific data from the user database.	Sh Interface [UMTS-SH]
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 [CHNG]
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 [DIAM]
NPI-14	A reference point from Charging FE and Authentication and Session Management FE.	Rf [CHNG]
NPI-15	This reference point controls the Resources and Admission Control.	Gq' [AFSPDF]
NPI-16	Reference point between the Transport Processing Function and Resource and Admission Control.	Re [RACS-RE]
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.	e4 [NASS-E4]
NPI-30	This reference point supports the IPTV Service Provider Discovery step of the service discovery procedure relying on IMS.	ISC interface [TS124503]
NPI-32	Reference point between the ASM FE and the IMS messaging AS. (This is the ISC interface defined by 3GPP in TS 23.228)	ISC interface [TS124503]
NPI-33	Reference point allowing interaction between IPTV Applications and the IPTV Metadata Control FE. This is not subject to standardization.	Not Specified
NPI-34	The reference point between the IMS messaging server and the notification services. It is based on IMS SIP as defined by 3GPP in TS 24.229.	SIP

NPI-36	This reference points 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 API (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 [OIPF_CSP2]
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 [OIPF_CSP2]
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, Program and Content Keys and ECM attached information are provided to the CoD Encryption Management Function.	HTTP
NPI-46	Reference point where the content Service, Program 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
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: 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.	

3.3 Interfaces to External Systems

3.3.1 Consumer Network

Table 4: External Interfaces from the Consumer Network

DLNA Function	Interface between the OITF and DLNA devices in the home.	DLNA

4 Structure of the document

Each section of this specification identified below defines the procedures that use a specific protocol:

Section 5: HTTP

Section 6: SIP and SIP/SDP

Section 7: RTSP

Section 8: IGMP and Multicast Protocol

Section 9: RTP/RTCP

Section 10: UPnP

Section 11: DLNA

Section 12: DHCP

Section 13: UDP

Section 14: FLUTE

Section 15: Diameter

The annexes cover the following topics:

Annex A: Change History (informative)

Annex B: Example Messages (informative)

Annex C: User Profile Description (informative)

Annex D: Mapping attributes for Scheduled Content

Annex E: <protocol> names

Annex F: System Infrastructure

Annex G: System Infrastructure Mechanisms (informative)

Annex H: Presence XML Schema

Annex I: Protocol Procedure Section Structure (informative)

Annex J: OITF-specific TR-135 and TR-106 Remote Management Objects

Annex K: New Event package for SIP SUBSCRIBE /NOTIFY (informative)

Annex L: Overview of Notification Services in OIPF R2 (informative)

Annex M: Fast Channel Change and Retransmission (FCC/RET)

Annex N: IG handling of IMPUs in association with GRUU (informative)

Annex O: FDT Schema Extensions

Annex P: IG Service Awareness

Annex Q: Definition of Content-Reporting Info Package

Annex R: Definition of Digital-Media-Purchase Info Package

Annex S: Definition of Parental-Control-Watched-Content Info Package

Annex T: Common Types

5 HTTP

5.1 HTTP Reference points

This section 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 sections, if no qualification is provided it must be understood that 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

5.2 IG as a protocol converter supporting the HNI-IGI interface – HTTP Option

In support of the HNI-IGI HTTP option the IG SHALL act 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 by [SIP]; 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 [TS124503], and stripping those headers before a request/response is sent to the OITF.
- The IG SHALL behave transparently to all SIP headers defined in [SIP], 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 sections 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.

5.3 Protocols for IPTV Service Functions

5.3.1 Multicast content streaming with SIP session management

5.3.1.1 Protocol over HNI-IGI – HTTP Option

When the OITF initiates, modifies or terminates a multcast content service, the OITF sends HNI-IGI messages containing the appropriate method, mapped to HNI-IGI as described in section 5.6.1, "OITF-IG Interface (HNI-IGI)".

The SIP-specific information in the related messages is described in section 6.1.2.1 , "Multicast content streaming with SIP session management." The SIP-specific information is mapped to the HNI-IGI protocol, as described in section 5.6.1, "OITF-IG Interface (HNI-IGI)". 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 [TS124503].

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 the following sections, if no qualification is provided, it MUST be understood that the interaction can be performed natively or as a DAE application.

5.3.1.1.1 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

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

HTTP Request Body: SDP offer containing the following elements (conforming to [TS183063]):

- 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 Annex D.1, "Mapping SDP attributes from DVB SD&S information."
  - 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 Annex D.1, "Mapping SDP attributes from DVB SD&S information."
- An a=bc_service: BCServiceId line SHALL indicate the multicast content service that the OITF intends to join (according to the mapping defined in Annex D.1, "Mapping SDP attributes from DVB SD&S information").
- 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 section 5.3.1.1.2, "Retrieval of bandwidth parameter for FCC and/or RET enabled multicast content service" 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 Annex D.1, “Mapping SDP attributes from DVB SD&S information.”
- A c-line according to the mapping defined in Annex D.1, “Mapping SDP attributes from DVB SD&S information.”

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 Annex D.1, “Mapping SDP attributes from DVB SD&S information.”
- A c-line according to the mapping defined in Annex D.1, “Mapping SDP attributes from DVB SD&S information.”
- 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 Annex D.1, “Mapping SDP attributes from DVB SD&S information.”).
- OPTIONALly one or more a=bc_service_package: <BCPackageId> as defined in Annex D.2, “Service Package SDP attributes.” 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>) which SHALL be set according to the mapping defined in Annex D.1, “Mapping SDP attributes from DVB SD&S information.”

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 sevice 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 RFC 2616 [HTTP], 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=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 RFC 2616 [HTTP]
- <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.	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Contact The URI parameter and the sip.instance feature tag MUST be included and MUST match what is sent in the contact header included in the registration request. The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP] (application/sdp)
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Supported	RFC 3261 [SIP] set to timer
X-OITF-Session-Expires	RFC 4028 [SES-TIMR]
X-OITF-Recv-Info	RFC 6086 [INFO-PKG]

SHALL be empty or the list of info packages the OITF is willing to receive	TS 26.237 [PSS-MBMS]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Session-Expires	RFC 4028 [SES-TIMR]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
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 section 12 of [PSS-MBMS] 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	RFC 6086 [INFO-PKG] TS 26.237 [PSS-MBMS]

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	RFC 3261 [SIP] ACK <Request URI> SIP/2.0

X-OITF-From	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact The URI parameter MUST be included, and MUST match what has been inserted in the INVITE message. The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]

5.3.1.1.2 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 section 5.3.2.1.1, “Retrieval of Session Parameters”, 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 section 5.3.1.1.1, “Session Initiation”) 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 RFC 4566 [SDP] expressed in kbps.

5.3.1.1.3 Session Modification

To join a service outside the set of channels negotiated at session initiation or to perform a bandwidth modification, the OITF SHALL 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.

5.3.1.1.4 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP], 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: The request URI MUST be set to the contact returned in the 200 OK for the invite.	RFC 3261 [SIP] BYE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.1.1.5 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.

5.3.1.1.6 Content Reporting and Management of Content Reporting of Watched Scheduled Content

5.3.1.1.6.1 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - see Table 10

HTTP Request Body: SHALL include the XML document as per Annex D of [PSS-MBMS]

The Content-Reporting Info Package SHALL contain an XML document defined in Annex D of [PSS-MBMS] 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
  - ServiceId – The BCServiceId of the channel
  - ProgrammeId – 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 RFC 2616 [HTTP].

5.3.1.1.6.2 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 RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <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 section 5.3.1.1.6.1, “Content Reporting by the OITF.”

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>.	RFC 3261 [SIP] INFO <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Contact The URI parameter MUST be included, and MUST match what was used in the SIP INVITE for the scheduled content session. The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]
X-OITF-Call-ID MUST match what was used in the SIP INVITE for the scheduled content session	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF_Info-Package SHALL be set to Content-Reporting Info Package	RFC 6086 [INFO-PKG]
X-OITF-Content-Type	RFC 6086 [INFO-PKG]

SHALL be set to "application/3gpp-ims-mbms-psscommand+xml" that corresponds to Annex D of [PSS-MBMS]	TS 26.237 [PSS-MBMS]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Content-Disposition	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

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 MUST match the contact URI included in the contact filed of the SIP INVITE for the scheduled content session	RFC 3261 [SIP] UPDATE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact The URI parameter and the sip.instance feature tag MUST be included and MUST match what is sent in the contact header included in the registration request.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 6086 [INFO-PKG]
X-OITF-Content-Length	RFC 3261 [SIP]
Recv-Info SHALL be set to remove support for the reception of the	RFC 6086 [INFO-PKG]

Content-Reporting Info Package OR SHALL be set to indicate support for the reception of Content-Reporting Info Package	

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.1.1.7 Network-based multicast content streaming Time Shift

5.3.1.1.7.1 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 section 5.3.1.1.1, “Session Initiation.”

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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 [RFC2837]. The first body part SHALL be an SDP offer containing the following elements conforming to [TS183063]:

- An m-line for the RTSP control channel set in accordance with section 5.3.2.1.2, “Session Initiation”
- An m-line for the unicast media delivery channel set in accordance with section 5.3.2.1.2, “Session Initiation”, and where the transport and codecs 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 [OFRANSR]. 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.
- BCServiceId 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 section 5.3.1.1.6, “Content Reporting and Management of Content Reporting of Watched Scheduled Content” 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 section 5.3.1.1.7.3, “XML Schema for OITF-IPTV Commands”, 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 RFC 2616 [HTTP]. The SDP answer SHALL be in accordance with section 6.1.2.2.2.5, "Protocol over NPI-26" 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 7

HTTP Request Body: Empty

5.3.1.1.7.2 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 section 5.3.1.1.7.1, "User-initiated Activation of multicast content streaming Time Shift."

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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 [RFC2837] 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 [OFRANSR]. 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.
- BCServiceId 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 section 5.3.1.1.7.3, "XML Schema for OITF-IPTV Commands", 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 RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 7

HTTP Request Body: Empty

5.3.1.1.7.3 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>
```

5.3.2 Unicast content streaming with SIP session management

5.3.2.1 Protocol for SIP session management over HNI-IGI – HTTP Option

5.3.2.1.1 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Includes the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP], as well as an SDP body containing the missing SDP parameters according to section 6.1.2.2.1.2, "Protocol over NPI-4, NPI-19, NPI-26."

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 Note: 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 clause 4.3.2.2 in [OIPF_META2] when CoD content identifiers are delivered via the Content Guide. For DAE applications signalling CoD, the wild card part is constructed according to clause 8.2.2 in [OIPF_DAE2]. • The domain part (<domain name>) is the IPTV Service Provider domain name, obtained from the IPTV Service Provider discovery function.	RFC 3261 [SIP] OPTIONS <Request URI> SIP/2.0

X-OITF-From	RFC 3261 [SIP]
X-OITF-To SHALL be set to the value of the request URI in the "X-OITF-Request-Line OPTIONS" header	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Accept	Set to application/sdp as per RFC 3261 [SIP]
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 section 12 of [PSS-MBMS]	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

5.3.2.1.2 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 [TS124503].

- SDP Parameters for the RTSP control channel

The RTSP content control media description SHALL be carried by TCP and follow [SDP-TCP]. Hence, the SDP parameters for the RTSP content control channel SHALL be set as follows:

- o 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 [SDP-TCP]. 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`)
- o An “a=setup” attribute SHALL be present and set to “active” as defined in [SDP-TCP]
  (ex. a=setup:active)
- o An “a= connection” attribute SHALL be present and set as “new” as defined in [SDP-TCP]
  (ex. a=connection:new)
- o 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.
  (ex. c=IN IP4 <IP_ADDRESS>)
- o 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:

- o 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>
- o <fmt> settings:
  - When MPEG2-Transport Stream [MPEG2TS] is used, <fmt> SHALL be “33” as specified in RFC 3551 [RFC3551]
  - When OPTIONAL Timestamped-TS defined by [DLNA] 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 [DLNA], 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
- o <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 kbps) 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 can not, or does not, wish to send Receiver Reports. This is the default setting, as explained in section 9.1.2.1, “Protocol over UNIT-17.” Note that if the OITF sends RTCP Receiver Reports, then these can be used as keep-alive messages, as shown in section 6.1.2.2.2.5, “Protocol over NPI-26.” 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 RFC 4756 [FEC]:

- 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 RFC 2616 [HTTP], and the same SDP answer body received by the IG, as described in section 6.1.2.2.2.5, “Protocol over NPI-26”.

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 RFC 2616 [HTTP]
- <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 sends 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 Note: The request URI MUST 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 clause 4.3.2.2 in [OIPF_META2] when CoD content identifiers are delivered via the Content Guide. For DAE applications signalling CoD, the wild card part is constructed according to clause 8.2.2 in [OIPF_DAE2]. • The domain part (<domain name>) is the IPTV Service Provider domain name, obtained from the IPTV Service Provider discovery function.	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To MUST be set to the value of the request URI in the “X-OITF-Request-Line INVITE” header	RFC 3261 [SIP]
X-OITF-Contact Notes: URI parameter SHALL be included and SHALL match what is sent in the contact header included in the registration request. Expires parameter SHOULD be included.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Supported	RFC 3261 [SIP] set to timer
X-OITF-Session-Expires	RFC 4028 [SES-TIMR]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]

X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact The URI parameter and the sip.instance feature tag MUST be included and MUST match what is sent in the contact header included in the registration request.	RFC 3261 [SIP]
X-OITF-Session-Expires	RFC 4028 [SES-TIMR]
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 section 12 of [PSS-MBMS]	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] ACK <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact The URI parameter SHALL be included, and SHALL match what has been inserted in the INVITE message. The IG	RFC 3261 [SIP]

includes all other mandatory parameters that are absent.	

5.3.2.1.3 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP].

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	RFC 3261 [SIP] BYE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]

X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.2.1.4 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

5.3.2.2 Protocol for streaming over UNIT-17

The use of the HTTP protocol on this reference point SHALL comply with [HTTP]

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.

5.3.3 Forced Play Out Control with SIP session management

5.3.3.1 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 section 5.3.2.1.2, “Session Initiation”.

5.3.4 Content Download

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.

5.3.4.1 Protocol over UNIT-17

The use of the HTTP protocol on this reference point SHALL comply with [HTTP].

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.

5.3.5 Purchase of Digital Media Service using SIP

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.

5.3.5.1 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 [TVA-MD]. 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.

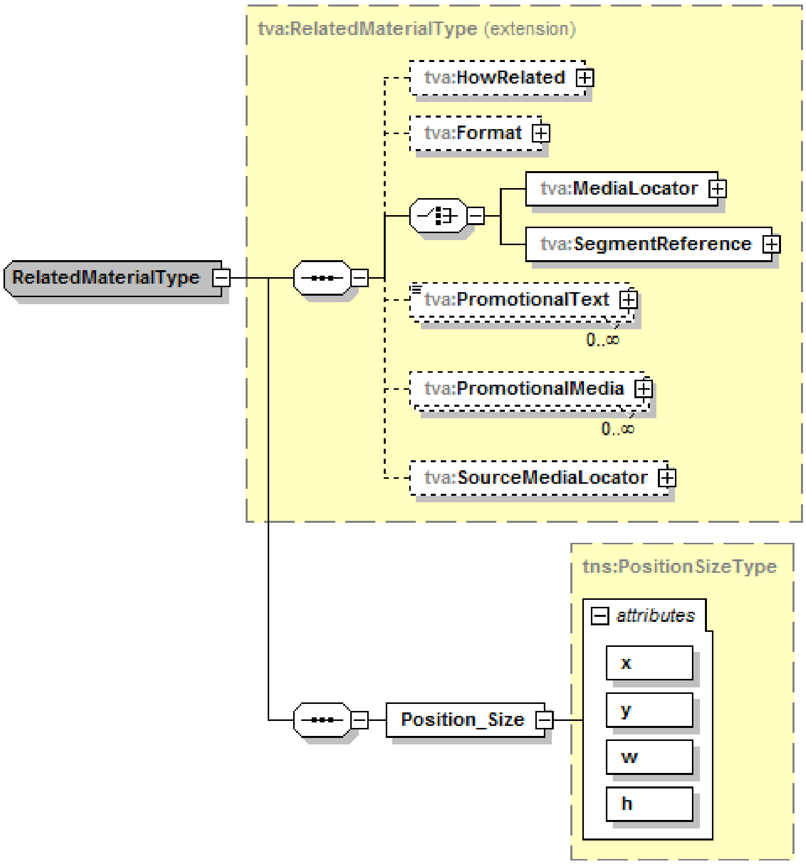


Figure 1: Elements of OIPF extended RelatedMaterialType

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, 1024×768), the OITF scales the Position_Size to fit in with that resolution.

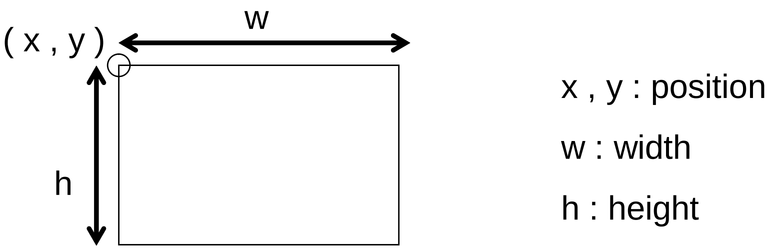


Figure 2: Construction of advertising window using (x,y,w,h) in Position_Size element

5.3.5.2 Advertising Digital Media through the use of overlays

The OITF advertise Digital Media to user 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.

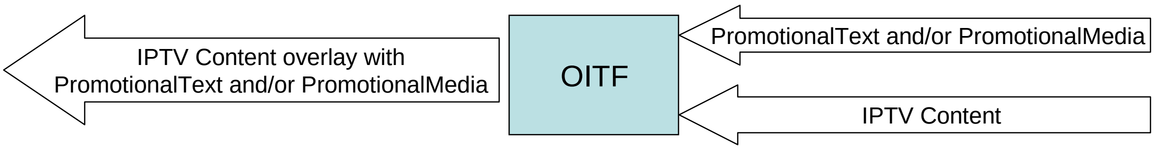

IPTV Content overlay with PromotionalText and/or PromotionalMedia
OITF
PromotionalText and/or PromotionalMedia
IPTV Content

Figure 3: Overlaying the content with advertisement for related media

5.3.5.3 Protocol over HNI-IGI – HTTP Option

5.3.5.3.1 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - see per Table 21

HTTP Request Body: (As defined in section 5.3.5.8, "XML Schema for Purchase Request of Digital Media", 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 RFC 2616 [HTTP]
- <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>.	RFC 3261 [SIP] INFO <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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"	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

X-OITF_Info-Package SHALL be set to “Digital Purchase”	RFC 6086 [INFO-PKG]
X-OITF-Content-Type SHALL be set to the “application/vnd.oipf.purchase+xml”	RFC 3261 [SIP]
X-OITF-Content-Disposition	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.5.3.2 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 RFC 2616 [HTTP].

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 RFC 2616 [HTTP].

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 MUST match the contact URI included in the contact field of the SIP INVITE	RFC 3261 [SIP] UPDATE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]

X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 6086 [INFO-PKG]
X-OITF-Content-Length	RFC 3261 [SIP]
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”	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.5.4 Protocol over UNIP-1

5.3.5.4.1 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 returns all stored purchased digital media records in an HTTP 200 OK response.

Note that the purchased digital media record can only updated by the IPTV Control FE, not the OITF.

5.3.5.5 Protocol over NPI-2

5.3.5.5.1 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.

5.3.5.6 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 section 5.3.5.7, "XML Schema for Advertisement of Digital Media".

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	RFC 2616 [HTTP] POST <Request URI> HTTP/1.1
Content-Length	RFC 2616 [HTTP]
Content-Type OPTIONAL. The default value of Content-Type is "text/plain; charset=us-ascii" as defined in RFC 2045 [RFC2045]	RFC 2616 [HTTP]
Host	RFC 2616 [HTTP]

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	RFC 2616 [HTTP] HTTP/1.1 <Status-Code> <Reason-Phrase>
Content-Length	RFC 2616 [HTTP]
Content-Type SHALL be set to "application/vnd.oipf.OIPFRelatedMaterial+xml"	RFC 2616 [HTTP]

5.3.5.7 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>
```

5.3.5.8 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.

5.3.6 Pay Per View multicast content service with SIP session management

5.3.6.1 Protocol over HNI-IGI – HTTP Option

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 section 5.6.1, “OITF-IG Interface (HNI-IGI).”

The SIP-specific information in the related messages is described in section 6.1.2.5, “Pay Per View multicast content service with SIP session management.” The SIP-specific information is mapped to the HNI-IGI protocol, as described in section 5.6.1, “OITF-IG Interface (HNI-IGI).” 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 [TS124503].

5.3.6.1.1 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 section 5.3.1.1.1, “Session Initiation”, 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 Annex D.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 Annex D.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 program belongs SHALL be included.
- An a=bc_program:BCprogramId line to indicate the PPV program 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.

5.3.6.1.2 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 section 5.3.1.1.3, “Session Modification.”

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 Annex M.2, “Fast Channel Change (FCC)”. Further, when multicast RET service is offered, the OITF SHALL join the RET stream as defined in Annex M.1, “Application Layer Retransmission (RET)”.

5.3.6.1.3 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 section 5.3.6.1.1, “PPV Service initiation without existing multicast content streaming session.”

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.

5.3.6.1.4 Session Termination

When the OITF wants to terminate the session, the OITF SHALL generate a HTTP POST request as described in section 5.3.1.1.4, “Session Termination” for an originating OITF.

5.3.7 Parental Control for Content using SIP

Note: Parental control relationship establishment, modification or deletion is out of scope of this specification.

5.3.7.1 Protocol for What is on TV – OITF initiated – HTTP Option

5.3.7.1.1 User 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP].

Step 4: Following that, the OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, “HNI-IGI Message Types”), 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 RFC 2616 [HTTP]. The body of the HTTP response SHALL include the SIP body received in the incoming NOTIFY (See also section 6.1.3.2.2, “Procedure for User Registration and Authentication in a network relying on IMS on UNIS-8.”)

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 RFC 2616 [HTTP]
- <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: The request URI can be set in one of 2 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)	RFC 3261 [SIP] and RFC 3265 [SIP-EVNT] SUBSCRIBE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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.	RFC 3261 [SIP]
X-OITF-Event SHALL be set to the Event Package Parental-Control-Watched-Content	RFC 3265 [SIP-EVNT] and Parental Control Watched Content (section 5.3.7.1.4, “XML Schema for Parental Control Watched Content”)
X-OITF-Accept SHALL be set to “application/vnd.oipf.iptvparentalcontrol.whatsontv+xml”	RFC 3265 [SIP-EVNT] and Parental Control Watched Content (section 5.3.7.1.4, “XML Schema for Parental Control Watched Content”)
X-OITF-Contact Notes: 1. The URI parameter and the sip.instance feature tag MUST be included and MUST match what is sent in the contact header included in the registration request. 2. Expires parameter SHOULD be included 3. Priority parameter SHOULD be included The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]

X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

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

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line Note: The Request URI MUST match the contact URI included in the contact field of the SIP SUBSCRIBE	RFC 3261 [SIP] NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3265 [SIP-EVNT] and Parental Control Watched Content (section 5.3.7.1.4, "XML Schema for Parental Control Watched Content")
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-Subscription-State	RFC 3265 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to "application/vnd.oipf.iptvparentalcontrol.whatsontv+xml"	RFC 3265 [SIP-EVNT] and Parental Control Watched Content (section 5.3.7.1.4, "XML Schema for Parental Control Watched Content")
X-OITF-Content-Length	RFC 3261 [SIP]

X-OITF-Contact	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

5.3.7.1.2 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.

5.3.7.1.3 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.

5.3.7.1.4 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 name="WhatsOnTvResponse" type="tns:WhatsOnTvResponseType"/>
  <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>
```

5.3.7.2 Protocol for Parental Control over HNI-IGI – HTTP Option

5.3.7.2.1 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 section 5.3.7.1, “Protocol for What is on TV – OITF initiated.” 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 section 6.1.2.6, “Parental Control for Content.” The SIP-specific information is mapped to the HNI-IGI protocol, as described in section 5.6.1, “OITF-IG Interface (HNI-IGI).”

Step 1: The OITF SHALL send an HTTP POST request to the IG using the HNI-IGI functionality, as described in section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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:oipf: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 section 5.3.7.2.3, "XML Schema for Parental Control."

The Content-Type of the message body SHALL be set to "application/vnd.oipf.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 RFC 2616 [HTTP].

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.	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From The From in the MESSAGE request SHALL be set to the identity of the controller.	RFC 3261 [SIP]
X-OITF-To The To in the MESSAGE request SHALL be set to the identity of the controlled user.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to "application/vnd.oipf.iptvparentalcontrol+xml"	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.7.2.2 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 a 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 section 8.1.1, “Multicast content streaming service on UNIS-13.” 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 section 5.3.2.1.3, “Session Termination”

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 RFC 2616 [HTTP]
- <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 MUST be set to the Public identity of the controlled user.	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From The From MUST be set to the Public identity of the	RFC 3261 [SIP]

controller.	
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”.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/vnd.oipf.iptvparentalcontrol+xml”	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

5.3.7.2.3 XML Schema for Parental Control

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:parentalcontrol:2011"
  xmlns:tns="urn:oipf:itpv: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:tPCCommand" />
      <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 controled 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="tPCCommand" 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>
```

5.3.8 Network-Based User Notification Services

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. This section 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 program starts at the IPTV SP discretion. 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.

5.3.8.1 Protocol over HNI-IGI – HTTP Option

5.3.8.1.1 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 34

HTTP Request Body: See section 5.3.8.6, “XML Schema for Notification Request for Broadcast Reminder.”

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 RFC 2616 [HTTP].

Step 4: Following that, the OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, “HNI-IGI Message Types”), 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: The request URI MUST be set to the well-known PSI for the notification service	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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"	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to "application/vnd.oipf.network-based-user-notification+xml"	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact The URI parameter and the sip.instance feature tag MUST be included and MUST match what is sent in the contact header included in the registration request.	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.8.2 Protocol over UNIS-6 for DAE-based Notification Management

5.3.8.2.1 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.

5.3.8.2.2 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 issue an XCAP GET to the user service profile and display 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.

5.3.8.2.3 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.

5.3.8.3 Protocol over UNIP-1

5.3.8.3.1 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).

5.3.8.3.2 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.

5.3.8.4 Protocol over NPI-2

5.3.8.4.1 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.

5.3.8.4.2 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.

5.3.8.5 Protocol over NPI-36

5.3.8.5.1 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.

5.3.8.6 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>
```

5.3.9 Content Bookmarking

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.

5.3.9.1 Protocol over HNI-IGI – HTTP Option

5.3.9.1.1 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 section 5.3.9.5, "XML Schema for Content Bookmarking".

Note that the SIP INFO including the CoD-Bookmark Info Package according to section 12 of [PSS-MBMS] 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 RFC 2616 [HTTP].

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.

5.3.9.1.2 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 section 12 of [PSS-MBMS] 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 RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - see Table 41

HTTP Request Body: Empty

5.3.9.1.3 Content-related bookmark retrieval over HNI-IGI

The procedure for retrieval of content related bookmarks is similar to the procedure defined in section 5.3.2.1.2, “Session Initiation” with the addition that the OITF has indicated its willingness to receive the CoD-Bookmark Info Package according to section 12 of [PSS-MBMS] 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 RFC 2616 [HTTP], and XML (see section 5.3.9.5, “XML Schema for Content Bookmarking”) in the SIP INFO body received by the IG , as described in section 6.1.2.8.3.1, “Session Initiation over UNIS-8”. 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - seeTable 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).	RFC 3261 [SIP] INFO <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/3gpp-ims-pss-mbms-bookmark+xml” that corresponds to Annex J of [PSS-MBMS]	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Recv-Info To indicate willingness to receive the CoD-Bookmark Info Package according to section 12 of [PSS-MBMS], 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	RFC 6086 [INFO-PKG]

CoD-Bookmark	

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
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.	RFC 3261 [SIP]

5.3.9.2 Protocol over UNIS-6 for DAE-based Content Bookmark 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 bookmark management is received from OITF, the IPTV Application FE manipulates the IPTV Service Profile FE.

5.3.9.2.1 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.

5.3.9.2.2 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.

5.3.9.2.3 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.

5.3.9.3 Protocol over NPI-2

5.3.9.3.1 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.

5.3.9.4 Protocol over UNIP-1

5.3.9.4.1 Protocol for Retrieving Stored Content Bookmarks

If the OITF desirers 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.

5.3.9.4.2 Protocol for Updating Content Bookmarks

If the OITF desirers 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.	RFC 3261 [SIP] INFO <Request URI> SIP/2.0

For a Scheduled Service: OIPF_IPTV_SC_Service@<domain name>. For a Content on Demand: OIPF_IPTV_CoD_Service_*@<domain name> .	
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Contact The URI parameter MUST be included, and MUST 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.	RFC 3261 [SIP]
X-OITF-Call-ID MUST match what was used in the SIP INVITE	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Info-Package SHALL be set to CoD-Bookmark according to section 12 of [PSS-MBMS].	RFC 6086 [INFO-PKG]
X-OITF-Content-Type SHALL be set to “application/3gpp-ims-pss-mbms-bookmark+xml” that corresponds to Annex J of [PSS-MBMS]	RFC 6086 [INFO-PKG], section 5.3.9.5, “XML Schema for Content Bookmarking”
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Content-Disposition	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]

X-OITF-CSeq	RFC 3261 [SIP]

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 MUST match the contact URI included in the contact filed of the SIP INVITE	RFC 3261 [SIP] UPDATE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 6086 [INFO-PKG]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Recv-Info To indicate willingness to receive the CoD-Bookmark Info Package according to section 12 of [PSS-MBMS], 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	RFC 6086 [INFO-PKG]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.3.9.5 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:bmk3gpp="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="bmk3gpp:BookmarkType">
      <xs:sequence>
        <xs:element name="Location" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </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 program 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 program 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)

5.3.10 Local PVR Service usig SIP

5.3.10.1 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 program to be recorded in the OITF
  - BCServiceId: identifies scheduled content channel from which the program 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) whom contents are recorded at. This parameter SHALL be shown as syntax of 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 RFC 2616 [HTTP].

Step 4: The OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, “HNI-IGI Message Types”) 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 RFC 2616 [HTTP]. 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 RFC 2616 [HTTP]
- <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 program to be recorded in the OITF
    - BCServiceId: identifies scheduled content channel from which the program 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 section 5.6.1.1, "HNI-IGI Message Types") 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 RFC 2616 [HTTP].

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 section 5.3.1.1.1, "Session Initiation."

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 Note: The request URI MUST be set to the well-known PSI for the PVR Service.	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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"	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]

SHALL be set to “application/vnd.oipf.pvr+xml”	
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact The URI parameter and the sip.instance feature tag MUST be included and MUST match what is sent in the contact header included in the registration request.	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

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 Note: The request URI MUST be set to the Public identity of the target of the message	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/vnd.oipf.pvr+xml”	RFC 3261 [SIP]
X-OITF-Accept-Contact	Set to the SIP Instance of the TargetDevice This parameter includes REQUIRED and explicated as RFC 3841
X-OITF-Content-Length	RFC 3261 [SIP]

5.3.10.2 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>
```

5.3.11 Network PVR (nPVR) using SIP

5.3.11.1 Protocol over HNI-IGI – HTTP Option

When the OITF initiates the request for network PVR Service Request, the procedure defined in section 5.3.10, “Local PVR” 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 program to be recorded in the OITF
- BCServiceId: Identifies the scheduled content channel from which the program 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 RFC 2616 [HTTP]
- <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 Note: The request URI MUST be set to the Public identity of the target of the message	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/vnd.oipf.pvrresult+xml”	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

5.3.11.2 XML schema for nPVR recording result

This section 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.

5.3.12 Personalised Channel

5.3.12.1 Procedure for Network-centric Personalized Channel (unicast only)

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

5.3.12.1.1 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 updates the PCh information towards the IPTV Service Profile FE through XCAP PUT over NPI-17, and then responds the OITF with a 200 OK.

5.3.12.1.2 PCh service provision

5.3.12.1.2.1 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 Set-up. The procedure is similar to the procedure described in section 5.3.2.1.2, “Session Initiation”, 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, section 5.3.12.1.1, “PCh profile configuration. “
- The domain part (<domain name>) is the IPTV Service Provider domain name, obtained from the IPTV Service Provider discovery function.

5.3.12.1.2.2 PCh session termination over HNI-IGI

The OITF SHALL send the request for a Personalised Channel session termination using the procedure as described in section 5.3.2.1.3, “Session Termination”.

5.3.12.2 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 which 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 [OIPF_PROTEX2].

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 section 5.3.10, “Local PVR.”

The messaging and procedures for recording the overlapped content item on an nPVR SHALL be as specified in section 5.3.11, “Network PVR (nPVR)”.

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.

5.3.13 Session Transfer with SIP session management

5.3.13.1 Protocol over HNI-IGI – HTTP Option

5.3.13.1.1 Generic Session Transfer Procedures

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

5.3.13.1.1.1 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 section 5.3.2.1.2, “Session Initiation”, with the following exceptions in 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 section 5.3.13.1.2.3, “Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode”.
- In the X-OITF Request Line, the wild card part (*) representing the content instance identifier to be transferred is constructed 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 section 5.3.13.1.2.3, “Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode”.

Furthermore, the IG handling in step 2 in section 5.3.2.1.2, “Session Initiation” is replaced by the IG handling as depicted in section 5.3.13.1.1.2, “IG handling of Session Initiation Requests related to a session transfer”.

The remaining steps in section 5.3.2.1.2, “Session Initiation” apply.

5.3.13.1.1.2 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 MUST 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 RFC 2616 [HTTP]
- <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 [TS124503] 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

5.3.13.1.2 Session Transfer via Push Mode

5.3.13.1.2.1 OITF Target Discovery

An OITF that wants to locate a target OITF for session transfer purposes SHALL perform the procedures described in section 5.4.6.1.4, “Procedure for Subscription to the Registration Event Package”.

Subsequently a target OITF can be selected from the returned information.

5.3.13.1.2.2 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 section 5.3.9.1.1, “IMS-based Content Bookmark Creation Request”. 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 46

HTTP Request Body: As per section 5.3.13.2, “XML Schema for Session Transfer Information included in a session transfer request from the transferor to transferee”.

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 RFC 2616 [HTTP].

Step 5: Following that, the transferor OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, "HNI-IGI Message Types"), 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 49

HTTP Request Body: Empty

Table 46: List 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)

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.	RFC 3261 [SIP] REFER <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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"	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
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 [RFC3891] • 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-	RFC 3261 [SIP] [SRVCONT] RFC 3891 [RFC3891]

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 OIPF. 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=AB03a0s09a2sdfglkj490333%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	
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/vnd.oipf.session-transfer+xml” corresponding to section 5.3.13.2, “XML Schema for Session Transfer Information included in a session transfer request from the transferor to transferee”	
X-OITF-Content-Length	RFC 3261 [SIP]

Table 47: List 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)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Response-Line	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

5.3.13.1.2.3 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 RFC 2616 [HTTP]. The body of the HTTP response SHALL include the XML structure as per section 5.3.13.2, "XML Schema for Session Transfer Information included in a session transfer request from the transferor to transferee".

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 section 5.3.13.1.1.1, "Transferee unicast session streaming Session Initiation associated with a 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 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 RFC 2616 [HTTP]
- <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 used to initiate a new session to handle the transfer. The OITF MAY follow section 5.3.2.1.1, "Retrieval of Session Parameters", if need be, towards the construction of the SDP.

Step 6: The transferee OITF SHALL then invoke the procedure defined in section 5.3.13.1.1.1, "Transferee unicast session streaming Session Initiation associated with a session Transfer".

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 section 5.6.1, "OITF-IG Interface (HNI-IGI)". The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP].

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

Table 48: 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 Notes: The Request URI MUST match the contact URI included in the contact field of the SIP REFER	RFC 3261 [SIP] NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3515 [RFC3515]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-Subscription-State	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “message/sipfrag”	RFC 3515 [RFC3515], RFC 3420 [RFC3420]
X-OITF-Length	RFC 3261 [SIP]

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

X-OITF HTTP Header	Source of Coding Information
X-OITF-Response-Line	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

5.3.13.2 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>
```

5.4 Protocol for Service Access and Control Functions

5.4.1 Service Provider Discovery

5.4.1.1 Protocol over HNI-IGI – HTTP Option

5.4.1.1.1 Retrieval of Service Provider Discovery Information

The procedures in this section 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 Annex C.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 RFC 2616 [HTTP].

Step 4: The OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, “HNI-IGI Message Types”), 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 a 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 RFC 2616 [HTTP]. 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 RFC 2616 [HTTP]
- <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 section 3.2.1 of [OIPF_META2].

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
  - o 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.
  - o 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)
  - o 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 TS 102 034 [TS102034] section 5.2.5, with the extended element defined in the Metadata Specification [OIPF_META2].

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 RFC 2616 [HTTP]
- <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.

5.4.1.1.2 Procedure for Cancellation of the Subscription

The procedure for de-registering the IPTV default user MUST 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.

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: 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>)	RFC 3261 [SIP] SUBSCRIBE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]

Note: The From user MUST be set to the IMPU of the default user.	
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”	RFC 3261 [SIP]
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”.	RFC 3265 [SIP-EVNT] and as per TS 183 063 [TS183063] section 5.1.2.2.1
X-OITF-Contact Notes: 1. URI parameter MUST be included, and MUST match the value that is sent in the Contact header in the registration request. 2. Expires parameter SHOULD be included 3. Priority parameter SHOULD be included The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires Note: If absent a default value according to RFC 3261 [SIP] SHALL be assumed by the IG To cancel the subscription, the X-OITF-Expires SHALL be set to 0	RFC 3261 [SIP]
X-OITF-Accept Set to “application/vnd.oipf.spdiscovery+xml”	RFC 3261 [SIP]
X-OITF-Content-Type OPTIONALly included when signalling OITF capabilities according schema defined in Annex C.2. SHALL be set to “application/vnd.oipf.ueprofile+xml”	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

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: The Request URI MUST match the contact URI included in the contact field of the SIP SUBSCRIBE	RFC 3261 [SIP], RFC 3265 [SIP-EVNT] and RFC 6080 [SIP-CFG] NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3265 [SIP-EVNT] and as per TS 183 063 [TS183063] section 5.2.2.2
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-Subscription-State	RFC 3265 [SIP-EVNT] and RFC 3856 [SIP-PRES]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/vnd.oipf.spdiscovery+xml”	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>

X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

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

5.4.1.1.3 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 is not refreshed.

5.4.1.2 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 TS 102 034 [TS102034] section 5.4.2 SHALL be used for the transport of IPTV Service Provider Discovery Information. The data delivered SHALL conform to TS 102 034 [TS102034] section 5.2.5, with the extension defined in [OIPF_META2].

When DAE applications are used, the HTTP protocol and data formats SHALL conform to section 5.3.1.1.6 of [OIPF_DAE2].

5.4.2 Service Discovery

5.4.2.1 Protocol over UNIS-6

The protocol on UNIS-6 SHALL be HTTP as defined in [OIPF_DAE2] 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.

5.4.2.2 Protocol over UNIS-15

The protocol used on UNIS-15 for the transport IPTV Service Discovery information SHALL be HTTP conforming to TS 102 034 [TS102034] section 5.4.2.

The IPTV Service Discovery information delivered via this protocol SHALL conform to TS 102 034 [TS102034] section 5.2.6 with the extension defined in [OIPF_META2]

5.4.3 Service Access

5.4.3.1 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 [OIPF_DAE2] for the details of the document format delivered via this protocol.

5.4.3.2 Protocol over UNIS-7

The use of the HTTP protocol on this reference point SHALL comply with section 4.1.2.2.2 (container based delivery) or section 4.2 (query mechanism) of the DVB-IP Broadband Content Guide specification [BCG].

The Content Guide metadata delivered via this protocol SHALL conform to TS 102 539 [BCG] with the extension defined in [OIPF_META2]

The OITF MAY request user specific information from the Metadata Control FE based on the IPTV Subscription Profile. (See section 5.4.4, "Subscription profile management and usage.")

5.4.4 Subscription profile management and usage

5.4.4.1 Protocols on UNIP-1 for XCAP-based Profile Management

The OITF SHALL be able to obtain a user's IPTV Subscription Profile. The format of the IPTV Subscription Profile SHALL conform to Annex C.1, "IPTV Subscription Profile." 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 RFC 4825 [XCAP].

UNIP-1 SHALL comply with XCAP as defined in RFC 4825 [XCAP].

5.4.4.1.1 XCAP Application Usage for IPTV Service

Profile Management

The XML Configuration Access Protocol (XCAP) defined in RFC 4825 [XCAP] 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.

XCAP application usage

XCAP requires application usages to fulfil a number of steps in the definition of such application usage. The remainder of this section 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 C.

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.

5.4.4.2 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.

5.4.4.3 Protocols over HNI-IGI – HTTP Option

5.4.4.3.1 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 section 5.6.1, “OITF-IG Interface (HNI-IGI)”. The content of the HTTP Request SHALL be as follows:

HTTP Request Header: includes the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP].

Step 4: The OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, "HNI-IGI Message Types"), 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 RFC 2616 [HTTP]. 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 RFC 2616 [HTTP]
- <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: The request URI MUST 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.	RFC 3261 [SIP], RFC 3265 [SIP-EVNT] and RFC 5875 [XCAP-EVT] SUBSCRIBE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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"	RFC 3261 [SIP]
X-OITF-Event The Event header SHALL be set to the "xcap-diff" event package.	RFC 3265 [SIP-EVNT] and as per TS 183 063 [TS183063] section 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	RFC 3265 [SIP-EVNT] and as per TS 183 063 [TS183063] section 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.	RFC 3265 [SIP-EVNT]

X-OITF-Contact Notes: The URI parameter SHALL be included and SHALL match what is sent in the Contact header included in the registration request The Expires parameter SHOULD be included	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires Note: If absent a default value SHALL be assumed by the IG	RFC 3261 [SIP]

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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

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: The Request URI MUST match the contact URI included in the contact field of the SIP SUBSCRIBE	RFC 3261 [SIP] NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3265 [SIP-EVNT] and as per TS 183 063 [TS183063] section 5.1.5.2
X-OITF-Call-ID	RFC 3261 [SIP]

X-OITF-Subscription-State	RFC 3265 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type SHALL be set to “application/xcap-diff+xml”.	RFC 3265 [SIP-EVNT] and as per TS 183 063 [TS183063] section 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

5.4.4.3.2 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 is not refreshed.

5.4.4.3.3 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.

5.4.5 Remote Management

5.4.5.1 General Procedures on UNI-RMS

The remote management functions REQUIRED for managed devices are specified in the general framework document TR069 [TR069] 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.

5.4.5.1.1 UNI-RMS for IG, AG and WAN Gateway

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

- TR-098 [TR098] that defines the data model for the “internet gateway device” SHALL apply to the WAN Gateway FE (see RMS3 functional block of the “Open IPTV Forum – Functional Architecture” document [OIPF_ARCH2])
- TR-106 [TR106] that defines the data model for the generic CWMP-managed device SHALL apply to the IG, AG and WAN-Gateway FEs.
- TR-104 [TR104] that defines the data model for the “SIP end-point” SHALL apply to the IG (see RMS2 functional block of the “Open IPTV Forum – Functional Architecture” document [OIPF_ARCH2]).

5.4.5.1.2 UNI-RMS for OITF

Although the remote management functions are specified in the general framework document TR-069 [TR069] 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 [TR069]. The limitations outlined in the following sections SHALL apply.

OITF RPC Methods Support Requirements

An OITF SHALL implement the following 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 doesn't support all the RPC requirements as defined in [TR069], 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 [TR069], with the following definition of the DeviceIdStruct that is used for the DeviceId argument of the Inform method:

> the 3 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 section 6.1.3.2.1, “User Identity Modelling.”)

- 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).

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)

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 [TR135] and TR-106 [TR106] 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 J, " OITF-specific TR-135 and TR-106 Remote Management Objects."

5.4.5.1.3 Configuration of the IG via Configuration File

CPE WAN Management protocol based on Broadband Forum TR-069 [TR069] 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 [OIPF_CSP2].
- 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.

5.4.5.1.3.1 Call Flow

There are 2 cases to be considered; the first case is when 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.

The second case is when 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.

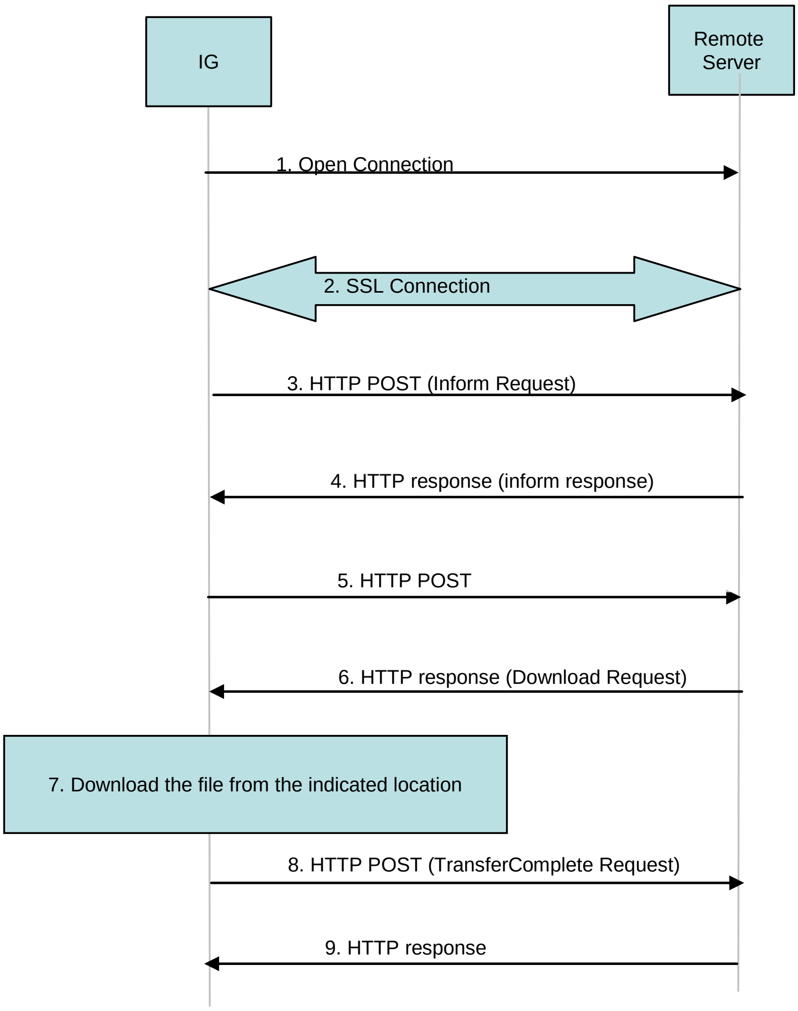

IG
Remote Server
1. Open Connection
2. SSL Connection
3. HTTP POST (Inform Request)
4. HTTP response (inform response)
5. HTTP POST
6. HTTP response (Download Request)
7. Download the file from the indicated location
8. HTTP POST (TransferComplete Request)
9. HTTP response

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, MUST be configured in the IG and remote server
Password:	OPTIONAL – If used, MUST 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 that the above sequence is an example and there are other valid sequences that can achieve the same result.

5.4.5.1.3.2 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 RFC 3261 (Section 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></Identifier>
    <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></Identifier>
    <Password>TheBoss</Password>
    <Alias>BruceSpringstein</Alias>
    <IMPI>household123@operator.com</IMPI>
    <SIPDigestPassword>CCXDFGGH</SIPDigestPassword>
  </AuthenticationSet>
  <GatewayAuthentication>true</GatewayAuthentication>
</IGconfiguration>
```

5.4.5.2 Remote Management using DAE APIs

See DAE Specification [OIPF_DAE2] section 7.11.

5.4.6 User Registration and Network Authentication

5.4.6.1 Procedure for User Registration and Authentication in network relying on IMS on the HNI-IGI Interface – HTTP Option

5.4.6.1.1 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <List of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 58

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 58. 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 59 in additional to the normal HTP headers as per RFC 2616 [HTTP].

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

5.4.6.1.2 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).

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.

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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 58

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 58. 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 59 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Table 58: List of 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 section 7.1.1 of TS 183 019 [TS183019]. The IG SHALL be responsible for resolving the domain name.	RFC 3261 [SIP] REGISTER <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Contact Notes: 1. Contact MUST include Feature Tags parameter. 2. URI parameter MUST be included. 3. Expires parameter SHOULD be included 4. Priority parameter SHOULD be included	RFC 3261 [SIP] and RFC 3840 [RFC3840]

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. 5. sip instance feature tag MUST be included according to sections 4.1 and 4.2 of [RFC5626]. Its format SHALL be identical to the format specified in [RFC4122]. The sip instance feature tag SHALL have the following syntax: +sip.instance="<urn:uuid:Unique-Instance>" where Unique-Instance SHALL be 128 bits and SHALL conform to the syntax in [RFC4122]. The Node field is a 48 bit field which SHALL be populated with the first 48 bits from the value employed in <deviceID> used at restart or powerup of an OITF device. The sip instance feature tag MUST 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"	
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Supported set to “gruu” if the feature is REQUIRED	RFC 3261 [SIP] RFC 5627 [RFC5627]

Table 59: List of 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	RFC 3261 [SIP] 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 RFC 5627 [RFC5627]

X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

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

5.4.6.1.3 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.

5.4.6.1.4 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 for the outgoing subscription request 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: 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 61 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Step 4: Following that, the OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, "HNI-IGI Message Types"), 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 62 in addition to the normal HTTP headers as per RFC 2616 [HTTP]. The body of the HTTP response SHALL include the SIP body received in the incoming NOTIFY (See also section 6.1.3.2.2, "Procedure for User Registration and Authentication in a network relying on IMS on UNIS-8.)"

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 63

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 60: 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: The request URI SHALL be set to the Public identity of the IPTV end user who has just registered	RFC 3261 [SIP] SUBSCRIBE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3265 [SIP] and RFC 3680 (registration event) [SIP-REG]
X-OITF-Accept	RFC 3265 [SIP-EVNT] and RFC 3680 [SIP-REG]
X-OITF-Contact Notes: 1. URI parameter SHALL be included, and SHALL match what is sent in the Contact header included in the registration request. 2. Expires parameter SHOULD be included 3. Priority parameter SHOULD be included The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]

Table 61: 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

Table 62: List of 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: The Request URI MUST match the contact URI included in the contact field of the SIP SUBSCRIBE	RFC 3261 [SIP] NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3265 [SIP-EVNT] and RFC 3680 [SIP-REG]
X-OITF-Call-ID	RFC 3265 [SIP-EVNT] and RFC 3680 [SIP-REG]
X-OITF-Subscription-State	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3265 [SIP-EVNT] and RFC 3680 [SIP-REG]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

Table 63: List of HTTP extension headers in the response to a NOTIFY request

X-OITF HTTP Header	Source of Information for Coding purposes
X-OITF-Response-Line	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

5.4.6.1.5 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 60 SHALL be set to 0.

For an OITF-initiated registration, the IG SHALL consider a subscription terminated if is not refreshed.

5.4.6.1.6 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.

5.4.6.1.7 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 [TS183063], [SMPL-IM], [TS124503], [RFC3840], [RFC3841]. 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.

5.4.6.2 GBA Authentication

Note that GBA authentication can be achieved using either the GBA Authentication using IMS Gateway procedure, specified in [OIPF_CSP2] section 5.4.5 or the, more general, procedure, HTTP Digest Authentication using IMS Gateway in [OIPF_CSP2] section 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 section describes the HNI-IGI message for the GBA Authentication. For the details of the sequence for GBA Authentication, refer to section 5.4.4 of [OIPF_CSP2]. Note that GBA authentication applies only for user registration and authentication based on IMS AKA.

5.4.6.2.1 Initial GBA registration

After IMS registration is successfully performed, and if the IG supports GBA Authentication, OITFs supporting native HNI-IGI SHALL issue 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 RFC 2616 [HTTP]
- <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.

5.4.6.2.2 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 [UB-UA]. The OITF SHALL act as the OIPF as specified in [UB-UA].

As a pre-requisite to this procedure, the GBA procedure MUST have been successfully completed.

The complete procedure for retrieval of credentials by the OITF from the IG is specified in [OIPF_CSP2].

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 RFC 2616 [HTTP]
- <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 [UB-UA].

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 [GAA] 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 [GAA]. The identifier for Ua security protocol HTTP Digest authentication according to 3GPP TS 33.220 [GAA] 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 RFC 2616 [HTTP]
- <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 [OIPF_CSP2].

HTTP Response Body: Empty

5.4.6.3 HTTP Digest Authentication

This section 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 section 5.4.4 of [OIPF_CSP2].

5.4.6.3.1 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 RFC 2616 [HTTP]

- <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 RFC 2616 [HTTP]
- <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 RFC 2617 [HTTPAUTH].
- <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

5.4.6.3.2 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 [OIPF_CSP2].

The complete procedure for use of HTTP credentials by the OITF retrieved from the IG is specified in [OIPF_CSP2].

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 RFC 2617 [HTTPAUTH]. The content of the HTTP Request SHALL be as follows:

HTTP Request Headers: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2617 [HTTPAUTH]
- <X-HNI-IGI- Auth-Realm> - set to realm as defined in RFC 2617 [HTTPAUTH]
- <X-HNI-IGI- Auth-Nonce> - set to nonce (server nonce) as defined in RFC 2617 [HTTPAUTH]
- <X-HNI-IGI- Auth-CNonce> - set to cnonce (client nonce) as defined in RFC 2617 [HTTPAUTH]
- <X-HNI-IGI- Auth-Algorithm> - set to algorithm as defined in RFC 2617 [HTTPAUTH]

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 RFC 2616 [HTTP]
- <X-HNI-IGI-Auth-Credentials> - this header is set to H(A1) as defined in RFC 2617 [HTTPAUTH]. If this header is empty or not present, there a 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 [OIPF_CSP2].

HTTP Response Body: Empty

5.4.6.4 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 RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP]

HTTP Response Body: a list of IMPUs and Display names (alias). Elements are separated with commas, entries are separates 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.

5.5 Protocols for Communications Functions using SIP

5.5.1 CallerID

5.5.1.1 Procedure for Instant Message Based Caller ID – HTTP Option

5.5.1.1.1 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 64. 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 65

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 section 5.3.2 of the DAE specification [OIPF_DAE2] titled “IMS Event Notification Framework”

Table 64: List of HTTP extension headers for an Instant Message Based Caller ID (IG→OITF)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line	RFC 3261 [SIP]

Note: The request URI MUST be set to the IMS Public Identity (IMPU) of the target of the message	MESSAGE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3428 [SIP-IM], Draft OMA-TS-SIMPLE_IM-V1_0-20080820-D [SMPL-IM]
X-OITF-Content-Length	RFC 3261 [SIP]

Table 65: List of 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.5.1.2 Procedure for IMS Telephony Based Caller ID (optional)

5.5.1.2.1 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 66. 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 RFC 3261 [SIP]. The content of the HTTP Request is as follows:

HTTP Request Header Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 67

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 68.

Note: For handling of new incoming INVITE messages for new dialogs, refer to section 5.3.2 of the DAE specification [OIPF_DAE2] entitled “IMS Event Notification Framework”

Table 66: List of 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: The request URI MUST be set to the IMS Public User Identity of the target of the message	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP], ES 283 002 [ES283002]
X-OITF-P-Called-Party-ID	ES 283 002 [ES283002]
X-OITF-P-Asserted-Identity	ES 283 002 [ES283002]

Table 67: List of 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Accept	RFC 3261 [SIP]

Table 68: List of HTTP headers in the HNI-IGI ACK Message (IG→OITF)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line	RFC 3261 [SIP] ACK <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Accept	RFC 3261 [SIP]

5.5.2 Instant Messaging

5.5.2.1 Procedure for Instant Messaging on HGI-INI – HTTP Option

Instant Messaging on the OITF uses the HNI-IGI functionality, as described in section 5.6.1, “OITF-IG Interface (HNI-IGI).”

There are two cases, messages originating from the OITF, and messages terminating in the OITF.

5.5.2.1.1 Procedure for OITF Originating an Instant Messaging

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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 69

HTTP Request Body: The content type as per RFC 3428 [SIP-IM]

Step 2: The IG SHALL validate that the request includes all the mandatory SIP headers for the message as per Table 69. 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 70 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Table 69: List of HTTP extension headers for an outgoing Instant Message (OITF→IG)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line Note: The request URI MUST be set to the Public identity of the target of the message	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0

X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

Table 70: List of 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.5.2.1.2 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 71. 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 70

HTTP Request Body: Empty

Step 4: The IG SHALL forward the SIP 200 OK to the network.

Table 71: 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: The request URI MUST be set to the Public identity of the target of the message	RFC 3261 [SIP] MESSAGE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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"	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP], Draft OMA-TS-SIMPLE_IM-V1_0-20080820-D [SMPL-IM]
X-OITF-Content-Length	RFC 3261 [SIP]

5.5.3 IM Session (Chat using MSRP)

5.5.3.1 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 72

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 72. 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

NOTE: 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 Annex B.2.1.2, "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 73, 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 MRSP 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 74

HTTP Request Body: Empty

Table 72: List of 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.	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To MUST be set to the value of the request URI in the "X-OITF-Request-Line INVITE" header	RFC 3261 [SIP]
X-OITF-Contact Notes: URI parameter SHALL be included and SHALL match what is sent in the Contact header included in the registration request. Expires parameter SHOULD be included	RFC 3261 [SIP]
X-OITF-Accept-Contact	Set according to [SMPL-IM]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Accept	[SMPL-IM]

SHALL be set to: “message/cpim”	

Table 73: List of 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Accept	RFC 3261 [SIP]

Table 74: 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 [SIP] ACK <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact The URI parameter MUST be included, and MUST match what has been inserted in the INVITE message. IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]

5.5.3.2 MSRP Invocation – HTTP Option

The OITF SHALL access MSRP capabilities in the IG using the X-HNI-IGI headers.

5.5.3.2.1 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of HNI-IGI headers encoded as HTTP headers> - as per Table 75

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 75. 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 76 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Table 75: List of 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	[SMPL-IM]
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 76: List of 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>	[SMPL-IM]
X-HNI-IGI-Message-ID	[SMPL-IM]
X-HNI-IGI-From	[SMPL-IM]
X-HNI-IGI-To	[SMPL-IM]

5.5.3.2.2 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of HNI- IGI headers encoded as HTTP headers> - as per Table 77

HTTP Request Body: SHALL contain the appropriate XML document as indicated in RFC 3994 [RFC3994] and OMA-TS-SIMPLE-IM_V1_0-20080820-D [SMPL-IM].

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, 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 76 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Table 77: List of 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 [SMPL-IM]
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	[SMPL-IM]
X-HNI-IGI-To	[SMPL-IM]

5.5.3.2.3 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The response SHALL include the HNI-IGI headers listed in Table 78, in addition to the mandatory HTTP headers in RFC 2616 [HTTP]. 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 RFC 2616 [HTTP]
- <list of HNI-IGI headers encoded in HTTP headers> - as per Table 79

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 78: List of 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	[SMPL-IM]
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 79: List of 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

5.5.3.2.4 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 80 in addition to the mandatory HTTP headers in RFC 2616 [HTTP]. The body of the HTTP 200 OK response SHALL contain the appropriate XML document as indicated in RFC 3994 [RFC3994] and OMA-TS-SIMPLE-IM_V1_0-20080820-D [SMPL-IM]

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 RFC 2616 [HTTP]
- <list of HNI-IGI headers encoded in HTTP headers> - as per Table 79

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: List of 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 [SMPL-IM]
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

5.5.3.3 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)." The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 81

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 81. 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 82 in addition to the normal HTTP 200 OK headers.

Table 81: List of HTTP extension headers for an MSRP BYE request (OITF→IG)

X-OITF HTTP Header	Source of Information for Coding purposes
X-OITF-Request-Line	RFC 3261 [SIP] BYE <Request URI> SIP/ 2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Length MUST be set to 0	RFC 3261 [SIP]

Table 82: List of 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	RFC 3261 [SIP] SIP/2.0 200 OK
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]

X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Content-Length Set to 0	RFC 3261 [SIP]

5.5.3.4 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 83, in addition to the mandatory HTTP headers in RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded in HTTP headers> - as per Table 84

HTTP Request Body : Empty

Step 4: The IG SHALL send SIP 200 OK to the network.

Table 83: List of HTTP extension headers for an Incoming SIP BYE (IG→OITF)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line Note: The Request URI MUST match the contact URI included in the contact field of the SIP INVITE (for outgoing session) or a 200 OK (for incoming session)	RFC 3261 [SIP] BYE <Request URI> SIP/ 2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Length MUST be set to 0	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

Table 84: List of HTTP extension headers for the response to an SIP BYE (OITF→IG)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Response-Line	RFC 3261 [SIP]

	SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.5.3.5 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 85. This is REQUIRED since the IG MUST 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 85, in addition to the mandatory HTTP headers in RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded in HTTP headers> - as per Table 86

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 87.

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 85: List of 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	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To MUST be set to the value of the request URI in the “X-OITF-Request-Line INVITE” header	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]
X-OITF-Accept-Contact	Set according to [SMPL-IM]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Accept SHALL be set to: “message/cpim”	[SMPL-IM]

Table 86: 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]

X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Accept SHALL be set to “message/cpim”	RFC 3261 [SIP]
X-OITF-Contact Notes: URI parameter SHALL be included and SHALL match what is returned in the contact header includes in the response to the registration process Expires parameter SHOULD be included	RFC 3261 [SIP]

Table 87: 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	RFC 3261 [SIP] ACK <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact The URI parameter SHALL be included, and SHALL match what been received in the incoming INVITE message.	RFC 3261 [SIP]

5.5.4 Presence

5.5.4.1 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 88

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 88. 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 89, in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Step 4: The OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, “HNI-IGI Message Types”), 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 90 in addition to the normal HTTP headers as per RFC 2616 [HTTP]. The body of the HTTP response SHALL include the SIP body received in the incoming NOTIFY compliant to Annex E of [TS183063].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 91

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.

5.5.4.2 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 88 SHALL be set to 0.

Table 88: List of HTTP extension headers for a SUBSCRIBE Request (OITF→IG)

X-OITF HTTP Header	Source of Information for Coding purposes
X-OITF-Request-Line Note: The request URI SHALL be set to the Public identity of the IPTV end user who has just registered	RFC 3261 [SIP] SUBSCRIBE <Request URI> SIP/2.0
X-OITF-From	RFC 3621 [SIP]
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”	RFC 3621 [SIP]

X-OITF-Event	RFC 3265 [SIP-EVNT], OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-Accept	RFC 3265 [SIP-EVNT], OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-Contact Notes: 1. URI parameter MUST be included, and MUST match the Contact header included in the registration request. 2. Expires parameter SHOULD be included 3. Priority parameter SHOULD be included IG includes all other mandatory parameters that are absent.	RFC 3621 [SIP]
X-OITF-Call-ID	RFC 3621 [SIP]
X-OITF-CSeq	RFC 3621 [SIP]
X-OITF-Expires Note: If absent a default value SHALL be assumed by the IG	RFC 3621 [SIP]
X-OITF-Content-Type	RFC 3265 [SIP-EVNT], OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-Content-Length	RFC 3261 [SIP]

Table 89: List of 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP]
X-OITF-Contact	RFC 3621 [SIP]

Table 90: List of HTTP extension headers for a SIP NOTIFY (IG→OITF)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line	RFC 3621 [SIP]

Note: The Request URI MUST match the contact URI included in the contact field of the SIP SUBSCRIBE	NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3621 [SIP]
X-OITF-To	RFC 3621 [SIP]
X-OITF-Event	RFC 3265 [SIP-EVNT], OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-Call-ID	RFC 3621 [SIP]
X-OITF-Subscription-State	RFC 3265 [SIP-EVNT],OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-CSeq	RFC 3621 [SIP]
X-OITF-Content-Type	RFC 3265 [SIP-EVNT] and OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-Content-Length	RFC 3261 [SIP]
X-OITF-Contact	RFC 3621 [SIP]

Table 91: 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	RFC 3621 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3621 [SIP]
X-OITF-To	RFC 3621 [SIP]
X-OITF-Call-ID	RFC 3621 [SIP]
X-OITF-CSeq	RFC 3621 [SIP]
X-OITF-Contact	RFC 3621 [SIP]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

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

5.5.4.3 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.

5.5.4.4 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 92

HTTP Request Body: As per section 5.5.4.6, “Presence Notification and Publish Schema.”

Step 2: The IG SHALL validate that the request includes all the mandatory SIP headers for the publication process as per Table 92. 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 93, in addition to the normal HTTP headers as per RFC 2616 [HTTP].

Table 92: List of HTTP extension headers for the PUBLISH Request (OITF→IG)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line Note: The request URI MUST be set to the IMS Public User Identity of the IPTV end user who has just registered	RFC 3261 [SIP] PUBLISH <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Event	RFC 3261 [SIP], OMA-ERP-Presence_SIMPLE-V1_1-20080627-A [SMPL-PRES]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Expires	RFC 3261 [SIP], RFC 3903 [RFC3903]
X-OITF-SIP-If-Match	RFC 3903 [RFC3903]
X-OITF-Content-Type	RFC 3261 [SIP]
X-OITF-Content-Length	RFC 3261 [SIP]

Table 93: List of HTTP extension headers for a response to SIP PUBLISH (IG→OITF)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Response-Line	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-ETag	RFC 3261 [SIP], RFC 3903 [RFC3903]
X-OITF-Expires	RFC 3261 [SIP]

5.5.4.5 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 RFC 3903 [RFC3903].

5.5.4.6 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 section 5.1.6 of [TS183063] “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 [SMPL-PRES]. New “service-id” values are defined for IPTV with the following values:

- IPTV-BC: Scheduled Content service, defined in [TS183063] section 5.1.6.
- IPTV-CoD: Content on Demand Service, defined in [TS183063] section 5.1.6.
- IPTV-NPVR: Network PVR Service, defined in [TS183063] section 5.1.6.
- Broadcast-TV: DVB-T/H/C/S Service, defined in Annex H, “Presence XML Schema”

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.

5.5.5 Content Sharing

5.5.5.1 Procedure for OITF Content Sharing Capability query

Capability of the other terminal can be queried, i.e. is recipient 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 94

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 94. 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 base on the black-white list. When the sender is in black list (means the sender has no right to query the receiving OITF's capability), the receiving OITF MUST reject the request with a non-200 OK response, including the reason for rejection. When the sender is in white list (means the sender has right to query the receiving OITF's capability), the receiving OITF MUST response 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 94

HTTP Request Body: application/sdp

The response body includes the SDP offer generated by the OITF. SDP SHALL be used as specified in [TS124503]

- An m=<media> <port> <transport> <fmt>
- An a=rtpmap:<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 [TS124503].

Table 94: List of 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	RFC 3261 [SIP] OPTIONS <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To MUST be set to the value of the request URI in the "X-OITF-Request-Line OPTIONS" header	RFC 3261 [SIP]
X-OITF-Contact Notes: URI parameter SHALL be included and SHALL match what is sent in the Contact header included in the registration request. Expires parameter SHOULD be included	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

X-OITF- Accept Accept: application/sdp	RFC 3261 [SIP]
X-OITF-Accept-Contact 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".	[VIDEOSHARE]
X-OITF-Content-Length Set to 0	RFC 3261 [SIP]

5.5.5.2 Invitation Procedure for OITF Originating a Content Sharing

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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 95

HTTP Request Body: application/sdp

The request body includes the SDP offer generated by the OITF. SDP SHALL be used as specified in [TS124503].

- 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=fmtp 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 95. 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 INVIE 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 96 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 97

HTTP Request Body: Empty

Table 95: List of 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.	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To MUST be set to the value of the request URI in the "X-OITF-Request-Line INVITE" header	RFC 3261 [SIP]
X-OITF-Contact Notes: URI parameter SHALL be included and SHALL match what is sent in the Contact header included in the registration request. Expires parameter SHOULD be included	RFC 3261 [SIP]
X-OITF-Accept-Contact 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".	[VIDEOSHARE]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

Table 96: List of 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	RFC 3261 [SIP] SIP/2.0 <response>
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact	RFC 3261 [SIP]

Table 97: 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 [SIP] ACK <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Contact The URI parameter MUST be included, and MUST match what has been inserted in the INVITE message. IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]

5.5.5.3 Procedure for OITF Refresh 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.

5.5.5.4 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 95.
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.

5.5.5.5 Transferring a Content Sharing Session

5.5.5.5.1 OITF Target Discovery

An OITF that wants to locate a target OITF for session transfer purposes SHALL perform the procedures described in section 5.4.6.1.4, “Procedure for Subscription to the Registration Event Package”.

Subsequently a target OITF can be selected from the returned information.

5.5.5.5.2 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, 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 section 5.6.1, “OITF-IG Interface (HNI-IGI)”. The content of the HTTP Request SHALL be as follows:

HTTP Request Header including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 98

HTTP Request Body: Empty

Step 3: The IG SHALL validate that the request includes all the mandatory SIP headers as per Table 98 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 RFC 2616 [HTTP].

Step 5: Following that, the transferor OITF SHALL send an HTTP HNI-IGI PENDING_IG request (refer to section 5.6.1.1, “HNI-IGI Message Types”), 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 99 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 49

HTTP Request Body: Empty

Table 98: List 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)

X-OITF HTTP Header	Source of Coding Information
X-OITF-Request-Line Note: The request URI SHALL be set to the identifier of the transferee (target OITF).	RFC 3261 [SIP] REFER <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
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”	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]

X-OITF-CSeq	RFC 3261 [SIP]
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 Content sharing session as per [RFC3891] • 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 OIPF. The media line for the media to be transferred SHALL include a port number with non zero value.	RFC 3261 [SIP] [RFC3515] [RFC3891]
X-OITF-Contact	RFC 3261 [SIP]

Table 99: 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: The Request URI MUST match the contact URI included in the contact field of the SIP REFER	RFC 3261 [SIP] NOTIFY <Request URI> SIP/2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Event	RFC 3515 [RFC3515]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-Subscription-State	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Length	RFC 3261 [SIP]

5.5.5.5.3 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 98, in addition to the normal HTTP headers as per RFC 2616 [HTTP]. The body of the HTTP response SHALL include the XML structure as per section 5.3.13.2, "XML Schema for Session Transfer Information included in a session transfer request from the transferor to transferee".

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 RFC 2616 [HTTP]
- <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 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 indicated 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 used 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 section 5.6.1, "OITF-IG Interface (HNI-IGI)".

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 95, requesting set up a new session with transferee OITF.

Step 7: The IG SHALL send a SIP INVIE 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 96 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 97

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 99 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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 49

HTTP Request Body: Empty

5.5.5.5.4 Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode

The remote OITF is having 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 a 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 95 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

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 96 in addition to the normal HTTP headers as per RFC 2616 [HTTP].

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 RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 97

HTTP Request Body: Empty

5.5.5.6 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).” The content of the HTTP Request SHALL be as follows:

HTTP Request Header: Including the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <list of SIP headers encoded as HTTP headers> - as per Table 100

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 100. 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 101 in addition to the normal HTTP 200 OK headers.

Table 100: List of HTTP extension headers for an SIP BYE request (OITF→IG)

X-OITF HTTP Header	Source of Information for Coding purposes
X-OITF-Request-Line Note: The request URI MUST be set to the contact return in the 200 OK for the invite.	RFC 3261 [SIP] BYE <Request URI> SIP/ 2.0
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
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	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]
X-OITF-Content-Length MUST be set to 0	

Table 101: List of 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	RFC 3261 [SIP] SIP/2.0 200 OK
X-OITF-From	RFC 3261 [SIP]
X-OITF-To	RFC 3261 [SIP]
X-OITF-Call-ID	RFC 3261 [SIP]
X-OITF-CSeq	RFC 3261 [SIP]

5.6 Protocols System Infrastructure Functions

5.6.1 OITF-IG Interface (HNI-IGI)

5.6.1.1 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 section 5.4.6.2.2, “Credential Retrieval by an OITF for Re-use of GBA Authentication”).

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>
```

The following table lists the HNI-IGI message types

Table 102: 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 MUST 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 MUST be a 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 section 5.6.1.6, “HNI-IGI Auxiliary Message”.)

5.6.1.2 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.

The following table lists header values for the HNI-IGI protocol that the IG and OITF SHALL support and lists the IG action on those headers.

Table 103: 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@oiptv.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 sections.

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 MUST be present.

5.6.1.3 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.

The following table 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 104: 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 and 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 MUST be present.

The IG handles the SIP state machines.

5.6.1.4 HNI-IGI PENDING_IG Message – HTTP Option

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 RFC 2616 [HTTP]
- <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 don’t include a SIP request or a SIP response are typically sent by applications in the OITF that don’t have any ongoing communication with a 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 don't include a SIP request or a SIP response SHALL be as follows:

HTTP Request Header: It includes the following:

- <list of HTTP headers> - as per RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP]
- <X-OITF Extension headers> Any number of those extension headers depending on the application

HTTP Response Body: <Appropriate Message Body if applicable>

5.6.1.4.1 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.

5.6.1.4.2 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 has been received is explicitly disconnected or timed out
- The IG received from the OITF application with an outstanding HNI-IGI PENDING_IG Request an HNI IGI SIP Request to terminate the session (a SIP BYE)
- The IG received from the OITF application with an outstanding HNI-IGI PENDING_IG Request an HNI IGI SIP Request to terminate an ongoing subscription (a SIP SUBSCRIBE with X-OITF-Expiry set to 0)

For the last 2 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.

5.6.1.5 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 RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP]
- <X-OITF Extension headers> Any number of those extension headers depending on the application

HTTP Response Body: <SIP Message Body if applicable>

5.6.1.6 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 RFC 2616 [HTTP]
- <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 RFC 2616 [HTTP]
- <X-HNI-IGI Extension headers> - Any number of those extension headers depending on the request

HTTP Response Body: <Application Message body of 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.

5.6.1.7 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.

5.6.1.8 Guidelines for Applications using the HNI-IGI interface – HTTP Option

This section 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 specification.
- 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 section 5.6.1.4.1, “Refreshing of HNI-IGI PENDING_IG Message.”
- 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 a SIP request is the case where the OITF application sends an ACK.

5.6.1.9 Error Recovery in the IG – HTTP Option

This section covers the handling in the IG for encountered error-cases:

1) 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 second before it responds to the network with a 487 or a 481 error message.

2) 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 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 section 5.4.6, “User Registration and Network Authentication” in case the OITF undergoes a restart.

3) 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.

5.7 Protocols for Content Preparation

5.7.1 Reference points NPI-45, NPI-46, NPI-CSPT3 and NPI-CSPG3

The reference points specified here are introduced in [OIPF_ARCH2].

5.7.1.1 Overview

This section 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 ⇔ SCS interface defined in [DVB-SC].

The interface described in this section 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 section 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.

5.7.1.2 Interface implementation

The interface is implemented as a WEB service. The associated Classification Scheme is provided in section 5.7.2.

Each reference point identifies a calling entity and a 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

Table 105: Reference Points calling entity and called entity

Three operations are defined. They are intended to be used according to the role of the functional entities as shown through Table 106**Error! Reference source not found.**.

Table 106: Operations according to functional entity roles

Key rotation handled by: **Key generation handled by:**	**Called entity**	**Calling entity**
Called entity	getKeyScheduleSignaling	setScheduleAndGetKeySignaling
Calling entity	This case is not supported.	setKeyScheduleAndGetSignaling

5.7.1.2.1 getKeyScheduleSignaling operation

5.7.1.2.1.1 Method and parameters

Table 107: getKeyScheduleSignaling method parameters

Operation name		**Description**
getKeyScheduleSignaling		Operation name
Field name	**Field type**	**Description**
GetKeyScheduleSignalingRequest	GetKeyScheduleSignalingRequestType	Input parameter
GetKeyScheduleSignalingResponse	GetKeyScheduleSignalingResponseType	Output parameter

5.7.1.2.1.2 Input parameter structure definition

Hereunder is the XML schema of the GetKeyScheduleSignalingRequest message:

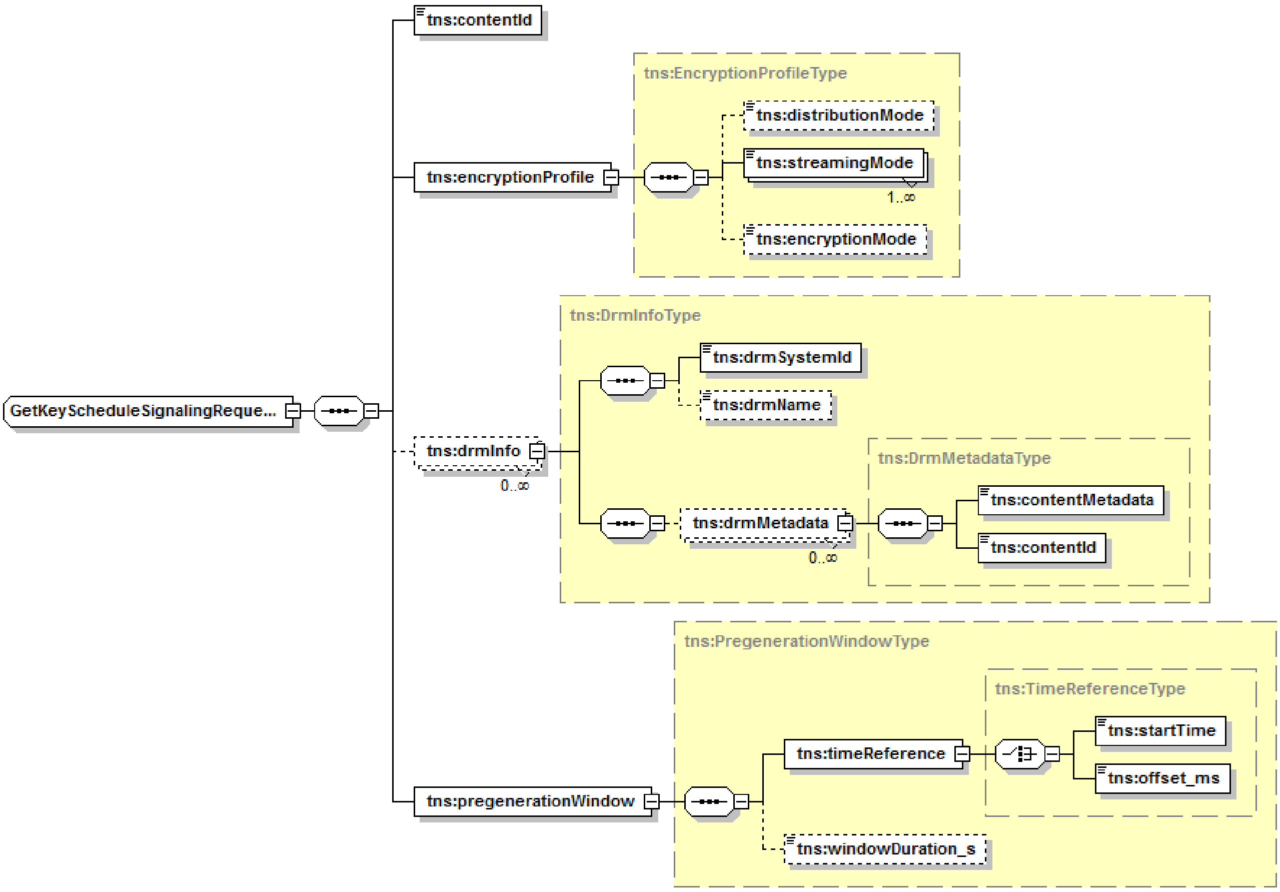


Figure 5: GetKeyScheduleSignalingRequest Schema

Table 108: 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.	5.7.1.2.5.6
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.	5.7.1.2.5.2
pregenerationWindow	PregenerationWindowType	1	The time information for the Key	5.7.1.2.5.8

			Management Function to know the period it is expected to generate keys for.	

5.7.1.2.1.3 Output parameter structure definition

Hereunder is the XML schema of the GetKeyScheduleSignalingResponse message:

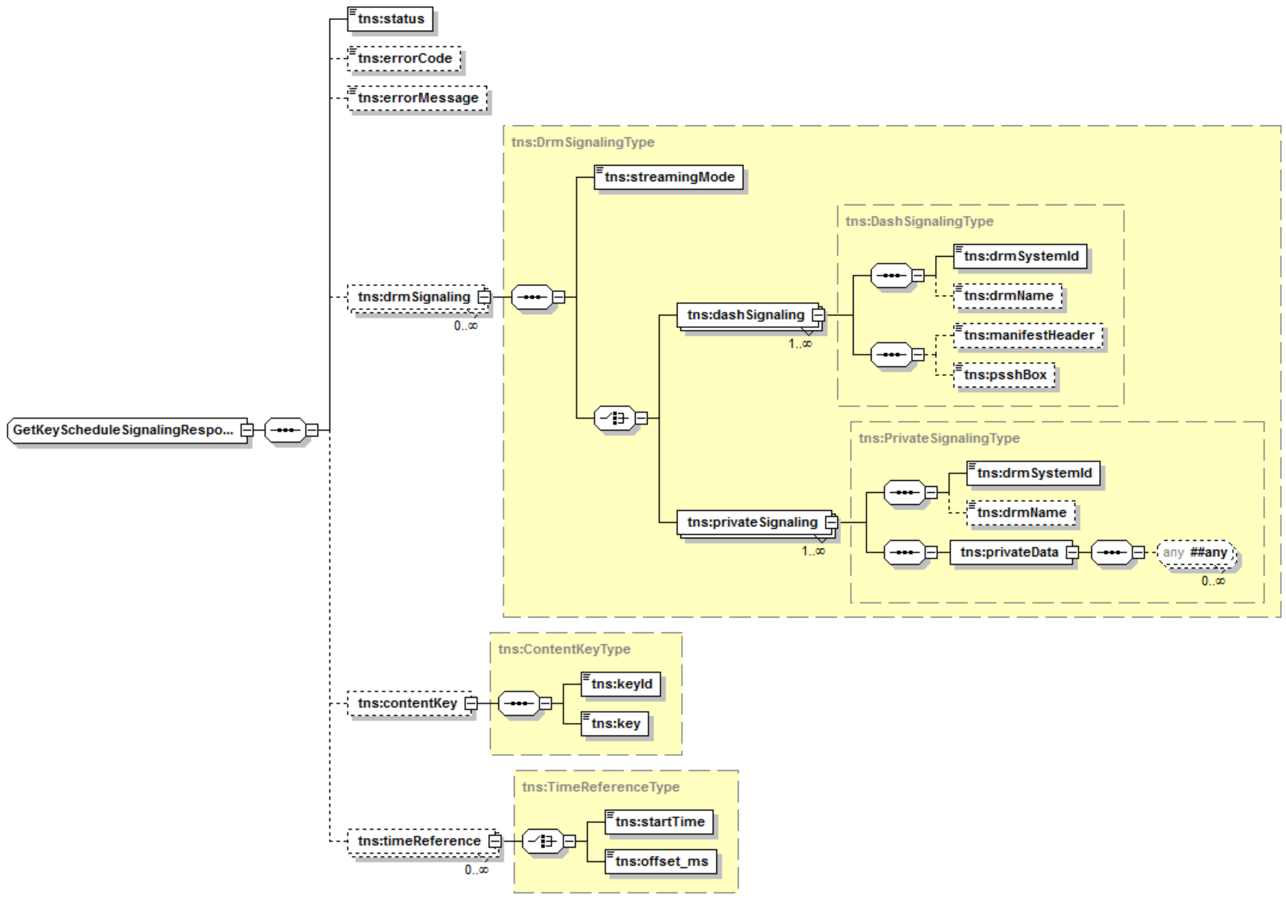


Figure 6: GetKeyScheduleSignalingResponse Schema

Table 109: GetKeyScheduleSignalingResponseType

Field name	Field type	Multiplicity	Description	Reference
status	StatusType	1	The global status of the request.	5.7.1.2.4.2
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 in case the request succeeds.	
errorMessage	xs:string	0..1	Any additional information related to the status in case of errors.	

			It is omitted in case the request succeeds.	
timeReference	TimeReferenceType	0..N	The dates and times for which keys have been generated. It is omitted in case the request failed.	5.7.1.2.5.12
contentKey	ContentKeyType	0..1	The content key to be used to protect the content. It is omitted in case the request failed.	5.7.1.2.5.1
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 in case the request failed.	5.7.1.2.5.4

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.

In case an error occurs, then no contentKey, no timeReference, and no drmSignaling elements will be provided.

5.7.1.2.2 setScheduleAndGetKeySignaling operation

5.7.1.2.2.1 Method and parameters

Table 110: setScheduleAndGetKeySignaling method parameters

Operation name		Description
setScheduleAndGetKeySignaling		Operation name
Field name	**Field type**	**Description**
SetScheduleAndGetKeySignalingRequest	SetScheduleAndGetKeySignalingRequestType	Input parameter
SetScheduleAndGetKeySignalingResponse	SetScheduleAndGetKeySignalingResponseType	Output parameter

5.7.1.2.2.2 Input parameter structure definition

Hereunder is the XML schema of the SetScheduleAndGetKeySignalingRequest message:

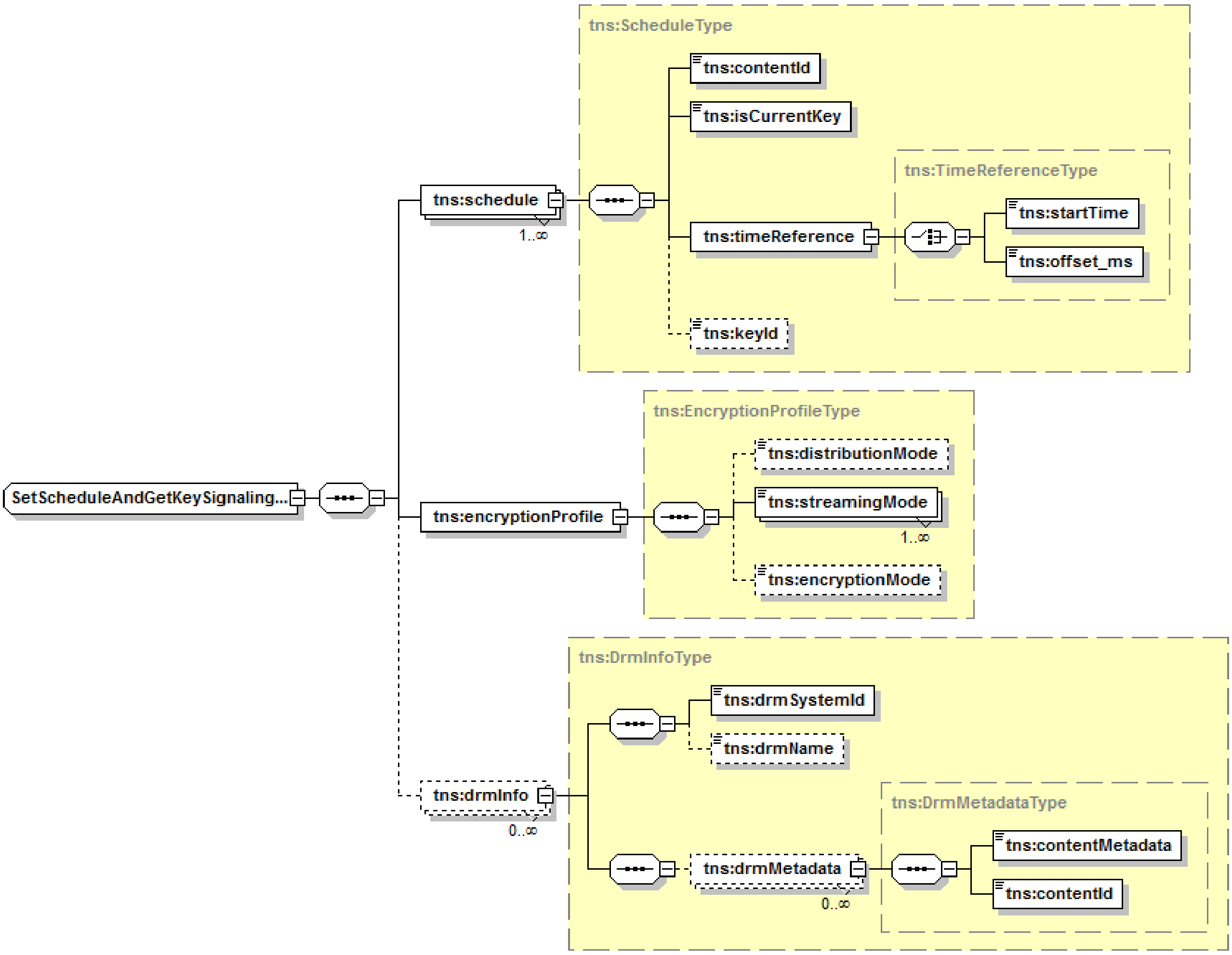


Figure 7: SetScheduleAndGetKeySignalingRequest Schema

Table 111: SetScheduleAndGetKeySignalingRequestType

Field name	Field type	Multiplicity	Description	Reference
encryptionProfile	EncryptionProfileType	1	Set of properties for the management of the content key and the signaling.	5.7.1.2.5.6
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.	5.7.1.2.5.2
schedule	ScheduleType	1..N	The times provided by the Encryption Function for which the Key Management Function has to generate keys.	5.7.1.2.5.11

5.7.1.2.2.3 Output parameter structure definition

Hereunder is the XML schema of the SetScheduleAndGetKeySignalingResponse message:

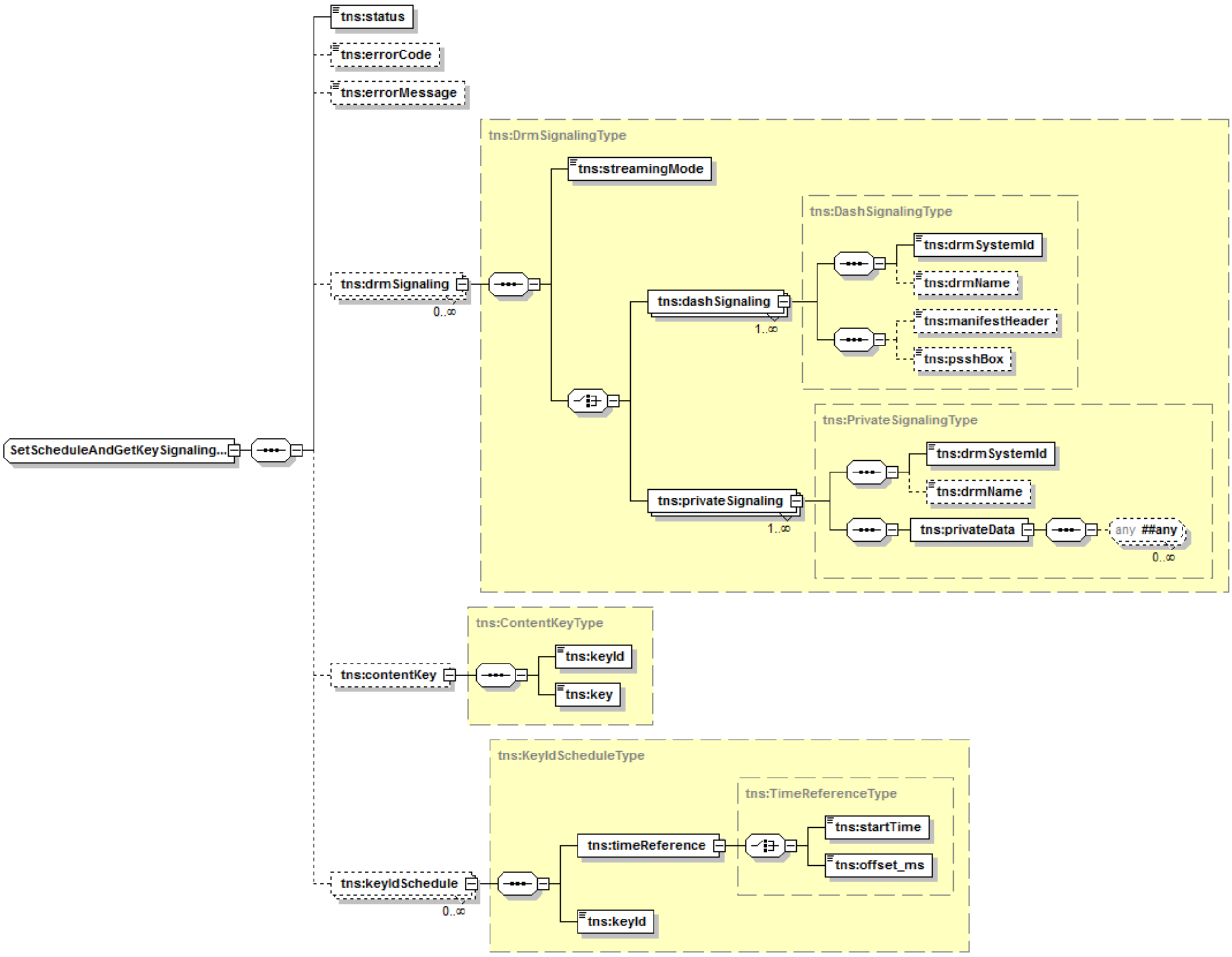


Figure 8: SetScheduleAndGetKeySignalingResponse Schema

Table 112: SetScheduleAndGetKeySignalingResponseType

Field name	Field type	Multiplicity	Description	Reference
status	StatusType	1	The global status of the request.	5.7.1.2.4.2
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 in case the request succeeds.	
errorMessage	xs:string	0..1	Any additional information related to the status in case of errors. It is omitted in case the request succeeds.	
contentKey	ContentKeyType	0..1	The content key to be used to protect the content. It is omitted in case the request failed.	5.7.1.2.5.1
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 in case the request failed.	5.7.1.2.5.4
keyIdSchedule	KeyIdScheduleType	0..N	The dates and times for which keys have been generated, and the keyId of these keys. It is omitted in case the	5.7.1.2.5.7

			request failed.	

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.

In case an error occurs, then no contentKey, no keyIdSchedule, and no drmSignaling elements will be provided.

5.7.1.2.3 setKeyScheduleAndGetSignaling operation

5.7.1.2.3.1 Method and parameters

Table 113: setKeyScheduleAndGetSignaling method parameters

Operation name		**Description**
setKeyScheduleAndGetSignaling		Operation name
Field name	**Field type**	**Description**
SetKeyScheduleAndGetSignalingRequest	SetKeyScheduleAndGetSignalingRequestType	Input parameter
SetKeyScheduleAndGetSignalingResponse	SetKeyScheduleAndGetSignalingResponseType	Output parameter

5.7.1.2.3.2 Input parameter structure definition

Hereunder is the XML schema of the SetKeyScheduleAndGetSignalingRequest message:

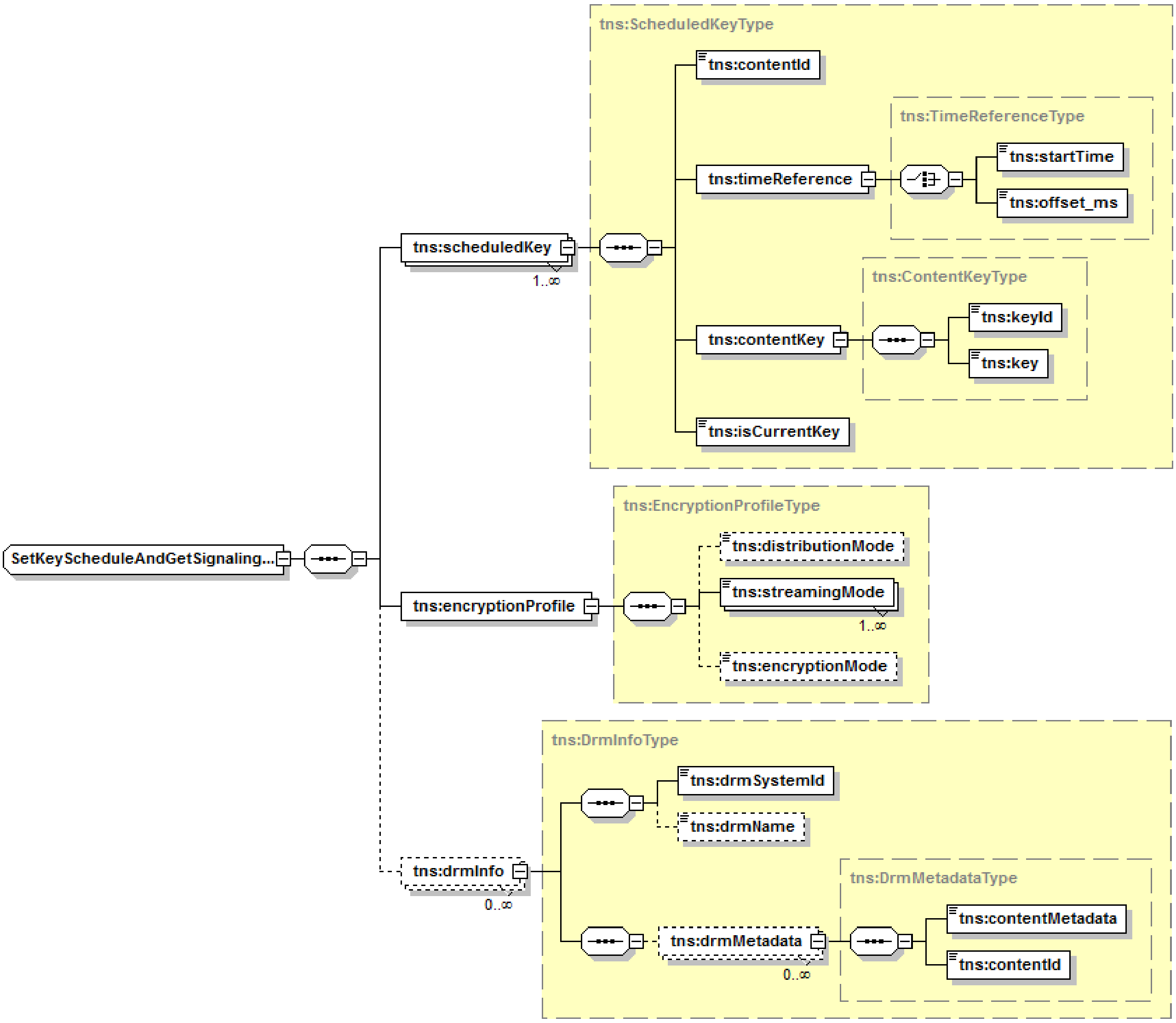


Figure 9: SetKeyScheduleAndGetSignalingRequest Schema

Table 114: SetKeyScheduleAndGetSignalingRequestType

Field name	Field type	Multiplicity	Description	Reference
encryptionProfile	EncryptionProfileType	1	Set of properties for the management of the content key and the signaling.	5.7.1.2.5.6
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.	5.7.1.2.5.2
scheduledKey	ScheduledKeyType	1..N	The keys provided by the Encryption Function.	5.7.1.2.5.10

5.7.1.2.3.3 Output parameter structure definition

Hereunder is the XML schema of the SetKeyScheduleAndGetSignalingResponse message:

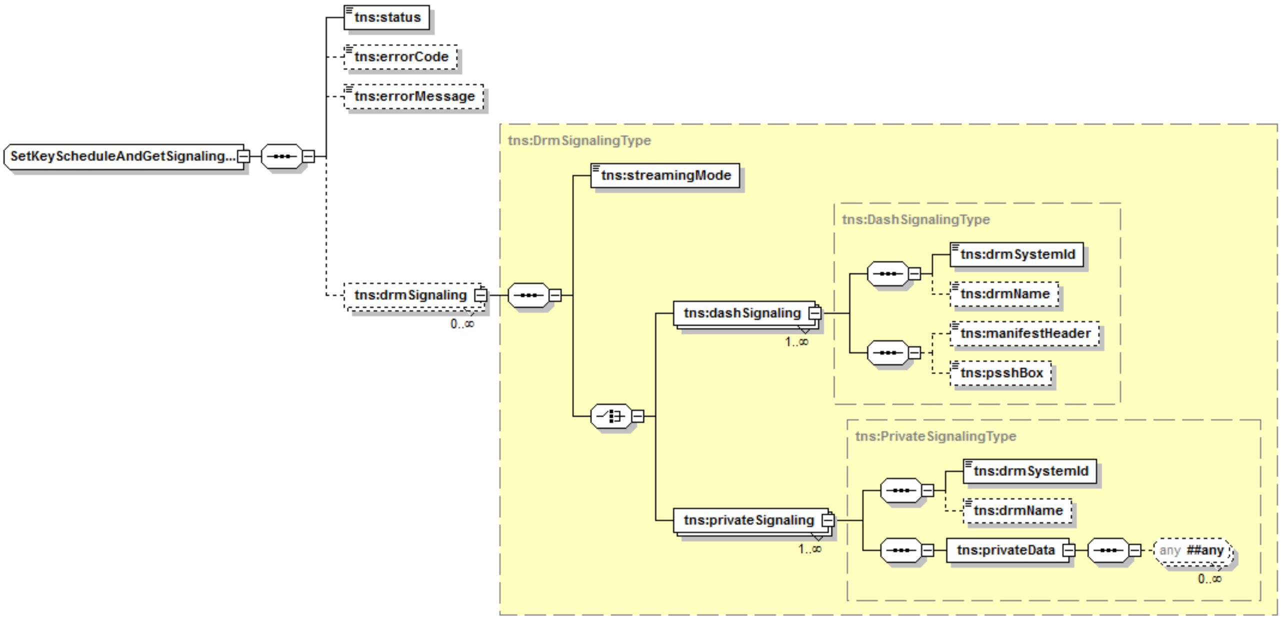


Figure 10: SetKeyScheduleAndGetSignalingResponse Schema

Table 115: SetKeyScheduleAndGetSignalingResponseType

Field name	Field type	Multiplicity	Description	Reference
status	StatusType	1	The global status of the request.	5.7.1.2.4.2
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 in case the request succeeds.	
errorMessage	xs:string	0..1	Any additional information related to the status in case of errors. It is omitted in case 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	5.7.1.2.5.4

			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 in case the request failed.	

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 everyaddressed 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.

In case an error occurs, then no drmSignaling element will be provided.

5.7.1.2.4 Simple types

5.7.1.2.4.1 UUIDType

This type is used for identifiers that need to be unique in time and space; it refers to [RFC4122].

Its main advantage is to not depend on centralized data storage.

Table 116: UUIDType

Base type	Pattern Restriction
xs:token	[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}

5.7.1.2.4.2 StatusType

Table 117: 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.

5.7.1.2.5 Data structures

5.7.1.2.5.1 ContentKeyType structure

Table 118: 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).

5.7.1.2.5.2 DrmInfoType structure

Table 119: 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.	5.7.1.2.5.3

5.7.1.2.5.3 DrmMetadataType structure

Table 120: 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.

5.7.1.2.5.4 DrmSignalingType structure

Either one of DASH or HAS field will be provided for a given DRM system.

Table 121: 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 section and/or a PSSH box data.	5.7.1.2.5.5
privateSignaling	PrivateSignalingType	0..N	Other signaling, e.g. OIPF HAS.	5.7.1.2.5.9

5.7.1.2.5.5 DashSignalingType structure

Table 122: 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 section to be inserted in the MPD.
psshBox	xs:base64binary	0..1	Pssh signaling information. DRM specific full pssh box binary block.

5.7.1.2.5.6 EncryptionProfileType structure

Table 123: 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.

5.7.1.2.5.7 KeyIdScheduleType structure

Table 124: KeyIdScheduleType structure

Field name	Field type	Multiplicity	Description	Reference
timeReference	TimeReferenceType	1	The date and time for which a key has been generated.	5.7.1.2.5.12
keyId	UUIDType	1	The unique identifier of the generated key, provided as a normalized UUID.	

5.7.1.2.5.8 PregenerationWindowType structure

Table 125: 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.	5.7.1.2.5.12
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.	

5.7.1.2.5.9 PrivateSignalingType structure

Table 126: 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.

5.7.1.2.5.10 ScheduledKeyType structure

Table 127: 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	5.7.1.2.5.12
contentKey	ContentKeyType	1	Content key value and identifier.	5.7.1.2.5.1

5.7.1.2.5.11 ScheduleType structure

Table 128: 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.	5.7.1.2.5.12
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.	

5.7.1.2.5.12 TimeReferenceType structure

Table 129: 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. It is in milliseconds.

5.7.2 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: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:UUIDType" 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 section 5.7.1.2.

```
<?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 name="body" element="kas:GetKeyScheduleSignalingRequest"/>
  </wsdl:message>
  <wsdl:message name="GetKeyScheduleSignalingResponse">
    <wsdl:part name="body" element="kas:GetKeyScheduleSignalingResponse"/>
  </wsdl:message>

  <wsdl:message name="SetKeyScheduleAndGetSignalingRequest">
    <wsdl:part name="body" element="kas:SetKeyScheduleAndGetSignalingRequest"/>
  </wsdl:message>
  <wsdl:message name="SetKeyScheduleAndGetSignalingResponse">
    <wsdl:part name="body" element="kas:SetKeyScheduleAndGetSignalingResponse"/>
  </wsdl:message>

  <wsdl:message name="SetScheduleAndGetKeySignalingRequest">
    <wsdl:part name="body" element="kas:SetScheduleAndGetKeySignalingRequest"/>
  </wsdl:message>
  <wsdl:message name="SetScheduleAndGetKeySignalingResponse">
    <wsdl:part name="body" 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>
```

6 SIP and SIP/SDP

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.

The next sections discuss these interfaces in more detail. The specification is structured in 2 sub-sections; one sub-section covers the SIP/SDP interface in the residential LAN (bullet 1 above), while the other sub-section covers the SIP/SDP interfaces within the managed network relying on SIP (last 3 bullets above). Within each sub-section, the interface details are specified on a per service basis.

6.1 SIP/SDP Reference Points within the Provider Network

This section 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 section 6.2.

6.1.1 Generic handling for SIP Requests/Responses

6.1.1.1 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.

6.1.1.2 Handling of Incoming and Outgoing SIP Requests

The IG SHALL support the procedures specified in [TS124503] for originating and terminating sessions.

6.1.2 Protocols for IPTV Service Functions

6.1.2.1 Multicast content streaming with SIP session management

6.1.2.1.1 Protocol over UNIS-8

6.1.2.1.1.1 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 section 5.3.1.1.2, “Retrieval of bandwidth parameter for FCC and/or RET enabled multicast content service”.

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.

6.1.2.1.1.2 Session Initiation and Modification

Upon receiving a request from the OITF for the initiation of a multicast content service session (see section 5.3.1.1.1, “Session Initiation”), the IG SHALL generate an initial INVITE request as specified in [TS124503] 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.

Session modification procedure is handled by the IG in the same way as a session initiation. See section 5.3.1.1.3, “Session Modification.”

6.1.2.1.1.3 Session Termination

On receiving a request from the OITF for the termination of a multicast content service session (see section 5.3.1.1.4, “Session Termination”), the IG SHALL generate a BYE request as specified in [TS124503] 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 section 5.6.1, “OITF-IG Interface (HNI-IGI)”). The behaviour of the UNIS-8 part of the IG SHALL comply with the procedure specified in [TS124503] for terminating UA.

6.1.2.1.1.4 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 section 5.3.1.1.6.1, “Content Reporting by the OITF”), the IG SHALL generate a SIP INFO including the Content-Reporting Info Package [INFO-PKG].

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 FEIPTV 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 FEIPTV Control FE.

Upon receipt of a 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.

6.1.2.1.1.5 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 [TS124503].
- 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.

6.1.2.1.2 Protocol over NPI-4

6.1.2.1.2.1 Retrieval of Bandwidth Parameter for FCC/RET enabled multicast content service

The SIP OPTIONS message SHALL conform to [TS124503] 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 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 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.

6.1.2.1.2.2 Session Initiation

The IPTV Control FE SHALL support the procedures specified in [TS183063] section 5.3.1.1.

The IPTV Control FE SHALL support the procedures specified in [TS124503] 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 [TS183063] section 5.3.1.1.

If the SDP parameters are validated successfully, the IPTV Control FE SHALL respond as defined in [TS183063] section 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.

6.1.2.1.2.3 Session Modification

The IPTV Control FE SHALL support the procedures specified in [TS183063] section 5.3.1.2. Network initiated session modification does not apply.

6.1.2.1.2.4 Session Termination

The IPTV Control FE SHALL support the procedures specified in [TS183063] section 5.3.1.4.

6.1.2.1.3 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 [PSS-MBMS], 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 a 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.

6.1.2.1.4 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 multiparts of the message body including, both, the SDP parameters, and the XML schema for the OITF-IPTV Commands per section 5.3.1.1.7.3, “XML Schema for OITF-IPTV Commands” 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 section 6.1.2.2.2, “Procedure for Unicast Service Session Initiation,” 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 section 6.1.2.2.2, “Procedure for Unicast Service Session Initiation” 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 section 6.1.2.2.3, “Session Termination.” Following the successful completion of the said procedure, the IPTV Control FE SHALL return a SIP 200 OK to the IG.

6.1.2.2 Unicast content streaming with SIP session management

6.1.2.2.1 Retrieving missing parameters in the SDP prior to session setup using SIP OPTIONS

6.1.2.2.1.1 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 section 5.3.2.1.1, “Retrieval of Session Parameters.”

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.

6.1.2.2.1.2 Protocol over NPI-4, NPI-19, NPI-26

The SIP OPTIONS message SHALL conform to [TS124503] 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 conten 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 TS 102 034 [TS102034].

The XML description that complies to TS 102 034 [TS102034] 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 for 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.

6.1.2.2.2 Procedure for Unicast Service Session Initiation

6.1.2.2.2.1 Session Initiation

The IG SHALL support the procedures specified in [TS124503] 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 [TS124503] (for an originating UA). See section 5.3.2.1.2, “Session Initiation”.

See example messages in Annex B.1.1, “Example Messages for unicast content streaming session setup with SIP session management.”

6.1.2.2.2.2 Protocol over NPI-4

The IPTV Control Function SHALL support the procedures specified in [TS124503] 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: 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.

6.1.2.2.2.3 Protocol over NPI-19

The CDN Controller FE SHALL support the procedures specified in [TS124503] 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

6.1.2.2.2.4 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.

6.1.2.2.2.5 Protocol over NPI-26

The Cluster Controller FE SHALL support the procedures specified in [TS124503] 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 RFC 3261 [SIP].

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 RFC 4145 [SDP-TCP]. 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.
  (ex: `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 RFC 4145 [SDP-TCP]. (ex: `a=setup:passive`)
- An “a= connection” attribute SHALL be present and set to “new” as defined in RFC 4145 [SDP-TCP]. (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 section 7.1.1.2.3, “RTSP Control for media delivery.”) If the timeout parameter is not specified, then a default value of 60 seconds SHALL be used (refer to section 12.37 of [RTSP])
      (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 section 6.2 of [RTP]).
  (ex. `b=AS:64`, indicating 64kbps)
- An a-line with a “sendonly”
  (ex. `a=sendonly`)
- A b-line, b=RR:<bandwidth-value>, specifying the agreed bandwidth value (in kbps) 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 kbps) 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.

6.1.2.2.3 Session Termination

Session termination for unicast content streaming SHALL follows the same SIP procedures as session termination for the multicast content streaming service. See section 6.1.2.1.1.3, “Session Termination.”

6.1.2.3 Forced Play Out Control with SIP session management

6.1.2.3.1 Forced Play Out Controlled by the Cluster Controller

Note: Client-based play out control is described in [OIPF_MEDIA2] sections 4.1 and 4.2, and in [OIPF_CSP2] section 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. The [OIPF_DAE2] specification 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.

6.1.2.3.1.1 Session Initiation

Forced Play Out Control session initiation over UNIS-8 is the same procedure as described in section 6.1.2.2.2.1, “Session Initiation”.

6.1.2.3.1.2 Protocol over NPI-4

The IPTV Control Function SHALL support the procedures specified in [TS124503] 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 section 5.3.2.1.2, “Session Initiation”. 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 section 6.1.2.3.1.6, “Forced Play Out Control Policy Attribute.”

Note: 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.

6.1.2.3.1.3 Protocol over NPI-19

Protocol for Forced Play Out Control over NPI-19 is the same as described in section 6.1.2.2.2.3, “Protocol over NPI-19”.

6.1.2.3.1.4 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, the IPTV Control Function changes the “Request-URI” accordingly.

6.1.2.3.1.5 Protocol over NPI-26

The protocol for Forced Play Out Control over NPI-26 is similar to that described in section 6.1.2.2.2.5, “Protocol over NPI-26.”

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.

6.1.2.3.1.6 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>}
```

6.1.2.4 Purchase of Digital Media Service using SIP

6.1.2.4.1 Purchase Request for Digital Media

6.1.2.4.1.1 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.

6.1.2.4.1.2 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 to section 5.3.5.8, “XML Schema for Purchase Request of Digital Media”, 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.

6.1.2.5 Pay Per View multicast content service with SIP session management

6.1.2.5.1 Protocol over UNIS-8

6.1.2.5.1.1 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 section 6.1.2.1.1.2, “Session Initiation and Modification.”

6.1.2.5.1.2 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 section 6.1.2.1.1.2, “Session Initiation and Modification.”

6.1.2.5.1.3 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 section 6.1.2.1.1.2, “Session Initiation and Modification.”

6.1.2.5.1.4 Session Termination

PPV multicast content streaming session termination over UNIS-8 SHALL be the same as Scheduled Content session termination as described in section 6.1.2.1.1.3, “Session Termination.”

6.1.2.5.2 Protocol over NPI-4

6.1.2.5.2.1 PPV service initiation without existing multicast content streaming session

The IPTV Control FE SHALL support the procedures specified in [TS183063] section 5.3.1.5.1.

The IPTV Control FE SHALL support the procedures specified in [TS124503] 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 section 6.1.2.1.2.2, “Session Initiation”, 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 program 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 program 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 program is allowed in the user profile, but has not started, i.e., the current time is earlier than the program start time, the IPTV Control FE SHALL NOT accept the offer and SHALL answer with a 403 error code.
    - If the program is allowed in the user profile, and has started, i.e., the current time is later than the program start time, the IPTV Control FE SHALL switch to the corresponding PPV service as defined in section 6.1.2.5.2.3 “Switching to a PPV service from a multicast content service or another PPV”.
- The IPTV Control FE SHALL process the a=bc_service_package attributes as defined in section 6.1.2.1.2.2, “Session Initiation.”

6.1.2.5.2.2 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 section 6.1.2.1.2.3, “Session Modification.”

6.1.2.5.2.3 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 ES 283 003 [ES283003] 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 section 6.1.2.5.2.1, “PPV service initiation without existing multicast content streaming session.” 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.

6.1.2.5.2.4 Session Termination

Upon receipt of a PPV session termination request, the IPTV Control FE SHALL follow the procedures defined in [TS124503] concerning the AS acting as a terminating UA, which is the same as multicast content streaming session termination as described in section 6.1.2.1.2.4, “Session Termination.”

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 [TS124503] for an originating OIPF, which is the same as multicast content streaming session termination as described in section 6.1.2.1.2.4, “Session Termination.”

6.1.2.6 Parental Control for Content using SIP

6.1.2.6.1 What is on TV – OITF initiated

6.1.2.6.1.1 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 section 5.3.7.1.4, “XML Schema for Parental Control Watched Content” SHALL be proxied to the OITF, otherwise it SHALL be rejected and no further processing SHALL be performed.
- The element ProgrammeId (section 5.3.7.1.4, “XML Schema for Parental Control Watched Content”) MAY be used.
- The OITF contact information SHALL be mapped to DeviceId.
- The OITF Call-ID SHALL be mapped to SessionId.

6.1.2.6.1.2 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 later 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 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 section 5.3.7.1.4, “XML Schema for Parental Control Watched Content”.

6.1.2.6.2 Parental Control

6.1.2.6.2.1 Protocol over UNIS-8

When receiving a request for Parental Control from the OITF as described in section 5.3.7.2.1, “Protocol for OITF Originating a Request for Parental Control,” the IG SHALL generate an initial MESSAGE request as specified in RFC 3428 [SIP-IM].

When receiving a request for Parental Control from the network as a SIP MESSAGE as described in section 5.3.7.2.2, “Protocol for OITF Receiving a Request for Parental Control,” 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.

6.1.2.6.2.2 Protocol over NPI-4

The IPTV Control Function SHALL support the procedures specified in [TS124503] as applicable to an AS acting as a SIP proxy or B2BUA.

Upon receipt of a 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.

6.1.2.7 Network-based User Notification Services using SIP

6.1.2.7.1 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 section 5.3.8.1.1, “Native HNI-IGI (IMS-based) Notification Request Setup Procedure”, the IG SHALL initiate a SIP MESSAGE which SHALL conform to [SMPL-IM]. The response to the SIP MESSAGE SHALL comply with [SMPL-IM]. The IG SHALL NOT retain any state information once the transaction is completed.

6.1.2.7.2 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.

6.1.2.8 Content Bookmarking using SIP

6.1.2.8.1 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 section 5.3.9.1.1, “IMS-based Content Bookmark Creation Request”), the IG SHALL generate a SIP INFO including the CoD-Bookmark Info Package according to section 2 of [PSS-MBMS].

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.

6.1.2.8.2 Protocol for Storing Content Bookmarks over NPI-4

Upon receipt of SIP INFO from the IG including the CoD-Bookmark Info Package according to section 2 of [PSS-MBMS] for storing a content bookmark, the IPTV Control FE SHALL examine the body.

If the body is not successfully validated as compliant to section 5.3.9.5, “XML Schema for Content Bookmarking”, 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 section 5.3.9.5, “XML Schema for Content Bookmarking”.

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.

6.1.2.8.3 Content-related bookmark retrieval

6.1.2.8.3.1 Session Initiation over UNIS-8

The IG SHALL support the procedures specified in [TS124503] 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 [TS124503] (for an originating UA). See section 5.3.2.1.2, “Session Initiation”.

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.

6.1.2.8.3.2 Session Initiation over NPI-4

The IPTV Control Function SHALL support the procedures specified in [TS124503] 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 replies 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 section 5.3.9.5, “XML Schema for Content Bookmarking”).

Note: 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.

6.1.2.8.3.3 Session Initiation over NPI-19

The procedure is the same as that defined in section 6.1.2.2.2.3, “Protocol over NPI-19”.

6.1.2.8.3.4 Session Initiation over NPI-25

The procedure is the same as that defined in section 6.1.2.2.2.4, “Protocol over NPI-25”.

6.1.2.8.3.5 Session Initiation over NPI-26

The procedure is the same as that defined in section 6.1.2.2.2.5, “Protocol over NPI-26”.

6.1.2.9 Local PVR Service using SIP

6.1.2.9.1 Protocol over UNIS-8

6.1.2.9.1.1 PVR Service Capture Request/Response

Upon receiving a request from the OITF for the PVR Service Capture Request (see section 5.3.10, “Local PVR”), the IG SHALL generate a SIP MESSAGE as specified in [TS124503].

6.1.2.9.1.2 PVR Record Request/Response

Upon receiving a request from the OITF for the PVR Service Capture Request (see section 5.3.10, “Local PVR”), the IG SHALL generate a SIP MESSAGE as specified in [TS124503].

The IG SHALL forward any received SIP response to the OITF including the 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.

6.1.2.9.2 Protocol over NPI-4

6.1.2.9.2.1 PVR Service Capture Request/Response

The IPTV Control Function SHALL support the procedures specified in [TS124503] 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.

6.1.2.9.2.2 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 [TS124503] 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 section 5.3.10.1.

The IPTV Control Function SHALL wait for the received response.

6.1.2.10 Network PVR (nPVR) using SIP

6.1.2.10.1 OITF-initiated nPVR

6.1.2.10.1.1 Protocol over UNIS-8

6.1.2.10.1.1.1 PVR Record Request/Response

Upon receiving a request from the OITF for the PVR Service Request (see section 5.3.11, “Network PVR (nPVR)”), the IG SHALL generate a SIP MESSAGE message as specified in [TS124503]. The IG SHALL forward any received SIP response to the OITF.

6.1.2.10.1.2 Protocol over NPI-4

The IPTV Control Function SHALL support the procedures specified in [TS124503] 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 program 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 section 5.3.11.2, “XML schema for nPVR recording result”.

At the start of the scheduled program 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 check whether the same program (identified by BCServiceID and ProgramID) has been requested to record. 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 program (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:

The Header: see Table 130.

The Body: the message body SHALL Multipart/Related MIME container [RFC2837], containing a 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 section 5.3.11.2, “XML schema for nPVR recording result”.

Table 130: List of SIP headers for PVR Service Recording Status (CC→IPTV Control)

SIP Header	Source of Coding Information
Request-Line Note: The request URI MUST be set to the SIP URI of the selected CDNC	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
From	RFC 3261 [SIP]
To The URI part of To SHALL be set to the value of the Request URI in the “Request-Line”	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Content-Type It SHALL be set to “multipart/related”	RFC 3261 [SIP]
Content-Length	RFC 3261 [SIP]

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 program identifier and the status information, then update the user profiles of all the users whom requested to record the program in the network. Then the IPTV Control FE SHALL generate a BYE request as specified in [TS124503] for originating sessions.

6.1.2.10.1.3 Protocol over NPI-19

The CDN Controller FE SHALL support the procedures specified in [TS124503] 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 program 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.

6.1.2.10.1.4 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.

6.1.2.10.1.5 Protocol over NPI-26

The Cluster Controller FE SHALL support the procedures specified in [TS124503] as applicable to a terminating UA.

Upon receiving a NPVR Request from the CDN Controller, the Cluster Controller SHALL examine the program 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 RFC 3261 [SIP].

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 to the CDF to record the content as defined in section 7.1.3.1, “RTSP Session Setup”.

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 section 5.3.11.2, “XML schema for nPVR recording result”.

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 131.
- The SIP UPDATE message Body: The Cluster Controller SHALL include a body associated with the appid “urn:oipf:service:PVR:2011” as defined in section 5.3.11.2, “XML schema for nPVR recording result”.

Table 131: List of SIP headers for PVR Service Recording Status (CC→IPTV Control)

SIP Header	Source of Coding Information
Request-Line Note: The request URI MUST be set to well-known PSI for NPVR	RFC 3261 [SIP] UPDATE <Request URI> SIP/2.0
From	RFC 3261 [SIP]
To The URI part of To SHALL be set to the value of the	RFC 3261 [SIP]

Request URI in the “Request-Line”	
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Content-Type It SHALL be set to “application/vnd.oipf.pvr+xml”	RFC 3261 [SIP]
Content-Length	RFC 3261 [SIP]

6.1.2.10.2 Non-OITF-initiated nPVR

nPVR service can also be initiated by non-OITF based user equipments so that the IPTV user can consume the multicast content later with an OITF. In this case, the user will send nPVR request to the IPTV Application (or other specific FE in the service provider domain) by means like 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 section 6.1.2.10.1, “OITF-initiated nPVR”.

6.1.2.10.2.1 Protocol over NPI-19

The protocol used in NPI-19 for Non-OITF initiate NPVR service SHALL be the same as for OITF Initiated Network PVR service, as defined in section 6.1.2.10.1.3, “Protocol over NPI-19”.

6.1.2.10.2.2 Protocol over NPI-25

The protocol used in NPI-25 for Non-OITF initiate NPVR service SHALL be the same as for OITF Initiated Network PVR service, as defined in section 6.1.2.10.1.4, “Protocol over NPI-25”.

6.1.2.10.2.3 Protocol over NPI-26

The protocol used in NPI-26 for Non-OITF initiate NPVR service SHALL be the same as for OITF Initiated Network PVR service, as defined in section 6.1.2.10.1.5, “Protocol over NPI-26”.

6.1.2.11 Personalised Channel using SIP

6.1.2.11.1 Procedure for Network-centric Personalised Channel (unicast only)

6.1.2.11.1.1 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 section 6.1.2.2, “Unicast content streaming with SIP session management”, 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.

6.1.2.11.1.2 Protocol over NPI-4

6.1.2.11.1.2.1 PCh session initiation

The IPTV Control Function SHALL support the procedures specified in [TS124503] 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: BCServiceId 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.

6.1.2.11.1.2.2 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: BCServiceId 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 RFC 3261 [SIP];
- The message body SHALL include one or more “PChContentSwitchControl” elements.

Note: The detailed XML schema refers to section 6.1.2.11.1.3, “XML schema for PCh content switch”.

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 section 7.1.4.2, “RTSP PLAY for PCh content switch.”

6.1.2.11.1.2.3 PCh session termination

Session termination for PCh SHALL follow the same SIP procedures as session termination for the COD service. See section 6.1.2.2.3, “Session Termination”.

6.1.2.11.1.3 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:itpv: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>
```

6.1.2.11.2 OITF-Centric Personalised Channel

6.1.2.11.2.1 Protocol over UNIS-8, NPI-4, NPI-19, NPI-25 and NPI-26

When recording overlapped content, the LPVR or nPVR procedures will be used.

The messaging and procedures for recording the overlapped content on a LPVR SHALL be as specified in section 6.1.2.8.3, “Content-related bookmark retrieval.”

The messaging and procedures for recording the overlapped content on a nPVR SHALL be as specified in section 6.1.2.10, “Network PVR (nPVR)”.

When a time gap between adjacent content items is detected, the OITF MAY perform the procedures specified in section 5.3.12.2, “OITF-centric Personalised Channel”.

6.1.2.12 Session Transfer with SIP session management

6.1.2.12.1 Protocol over UNIS-8

6.1.2.12.1.1 Generic Procedures

6.1.2.12.1.1.1 Session Initiation Procedures Related to a Session Transfer

The IG SHALL support the procedures specified in [SRVCONT] 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 [SRVCONT].

6.1.2.12.1.1.2 IG initiated Session Modification Procedures

The IG SHALL support the procedures specified in [TS124503] 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.

6.1.2.12.1.2 Session Transfer Via Push Mode

The following SHALL be supported by the IG:

- An outgoing REFER message associated with transferor OITF SHALL conform to [SRVCONT]
- The IG SHALL forward any incoming SIP REFER, for the purpose of session transfer, that conforms to the [SRVCONT] 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 [SRVCONT] and [RFC3515].
- The IG SHALL forward any incoming SIP NOTIFY, reporting the outcome of a session transfer associated with a REFER request, that conforms to [SRVCONT] and [RFC3515] to the transferor OITF. Non-conformant SIP REFER messages SHALL be rejected with the appropriate response code.

6.1.2.12.2 Protocol over NPI-4

6.1.2.12.2.1 Generic Procedures

6.1.2.12.2.1.1 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 [SRVCONT] 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 section 6.1.2.1.1.2, “Session Initiation and Modification”.

6.1.2.12.2.1.2 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.

6.1.2.12.2.2 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 [SRVCONT]. 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).

6.1.3 Protocols for Service Access and Control Functions

6.1.3.1 Service Provider Discovery

6.1.3.1.1 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 [TS183063], section 5.1.2.2.1. To wit:

- 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 values "TV-OITF", "STB-OITF", according to the implementation options in [OIPF_ARCH2] section 5.3.4.

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 RFC 3265 [SIP-EVNT].

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 [OIPF_META2].

Note: 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 K)

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.

6.1.3.2 User Registration and Network Authentication

6.1.3.2.1 User Identity Modelling

Every IMS Subscription SHALL be allocated a single unique default IMS Pubic 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.

This release complies with option 1 in Annex D.4 in [OIPF_ARCH2].

6.1.3.2.2 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 N 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 TS 124 503 [TS124503]. 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 TS 124 503 [TS124503]. 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 TS 124 503 [TS124503].
- For IG-initiated registrations, and following a successful registration process, the IG SHALL SUBSCRIBE to the registration event package in accordance with TS 124 503 [TS124503].
- 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 TS 124 503 [TS124503].
- 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 TS 124 503 [TS124503].
- For IG-initiated deregistration, and following the successful deregistration process, the IG SHALL terminate the subscription to the registration event package, as per TS 124 503 [TS124503].

The appropriate procedure (SIP Digest or IMS AKA) SHALL be followed by the IG for user registration and authentication.

6.1.3.3 Notification of Service Profile changes

6.1.3.3.1 Protocol over UNIS-8

6.1.3.3.1.1 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 RFC 5875 [XCAP-EVT] and RFC 5874 [XCAP-DFF].

The IG will process the request from the OITF and will generate a SUBSCRIBE request, that SHALL be as specified in [TS183063] section 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.

6.1.3.3.1.2 Processing of notifications

Refer to [TS183063] section 5.1.5.2

6.1.4 Protocols for Communication Services using SIP

6.1.4.1 CallerID

6.1.4.1.1 Procedures for Caller ID on UNIS-8

6.1.4.1.1.1 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 section 6.1.4.2.1, "Procedure for Instant Messaging on UNIS-8."

6.1.4.1.1.2 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.

6.1.4.2 Instant Messaging

6.1.4.2.1 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 MUST conform to OMA Instant Messaging using SIMPLE Draft OMA-TS-SIMPLE_IM-V1_0-20080820-D [SMPL-IM] to be acceptable for processing by the IG. Non-conformant SIP MESSAGE messages SHALL be rejected in accordance with [SMPL-IM] with the appropriate response code.
- For an incoming SIP MESSAGE to the IG that is conformant to [SMPL-IM], 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 [SMPL-IM]. The response to the SIP MESSAGE SHALL comply with [SMPL-IM].
- The IG SHALL NOT retain any state information once the transaction is completed.

6.1.4.3 Presence

6.1.4.3.1 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 [SMPL-PRES].
- An incoming NOTIFY messages that does not comply with [SMPL-PRES] SHALL be rejected with the appropriate error code in accordance with [SMPL-PRES], and no further processing SHALL be performed.
- On request from the OITF, the IG SHALL refresh the Presence subscription in accordance with [SMPL-PRES].
- 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.

6.1.4.3.2 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 [SMPL-PRES] for publishing presence information.

6.1.4.4 Chatting

6.1.4.4.1 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 [SMPL-IM].

The IG SHALL perform path mapping between Chatting peers as indicated in section 5.5.3, “IM Session (Chat using MSRP).”

The IG SHALL handle translation of Chat session initiation and teardown procedures when requested by the OITF as per section 5.4.5, “Remote Management.”

The IG SHALL handle translation of outgoing and incoming MSRP chat message as per section 5.5.3, “IM Session (Chat using MSRP).”

6.1.4.4.2 IM Session using MSRP over NPI–3

The P2P Chatting communication enablers FE SHALL confirm to the IM Server procedures described in OMA-TS-SIMPLE_IM-V1_0-20080820-D [SMPL-IM].

6.1.4.5 Content Sharing

6.1.4.5.1 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 [SIP] for Capability querying.

Incoming SIP INVITE 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 be acceptable for processing by the IG. Non-conformant SIP REFER SHALL be rejected in with the appropriate response code.

Incoming SIP UPDATE be acceptable for processing by the IG. Non-conformant SIP UPDATE SHALL be rejected in with the appropriate response code.

6.2 SIP/SDP Reference Points within the Residential Network

This section 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 section 5. 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 section 5 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.

6.2.1 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 [TS124503], 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 P 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 [SIP], 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 sections 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.

6.2.2 Protocols for IPTV Service Functions

6.2.2.1 Multicast content streaming with SIP session management

The IG SHALL support the procedures specified in [TS124503] for originating sessions.

6.2.2.1.1 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 [SIP] (e.g. Via, Max-Forwards). The OITF SHALL also include an SDP offer conformant to step 1 in section 5.3.1.1.1, “Session Initiation”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INVITE includes all the mandatory headers as per [SIP] and the headers listed in Table 5. Following a successful validation, the IG SHALL generate an initial SIP INVITE request as specified in [TS124503] 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 [SIP].

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 [SIP].

6.2.2.1.2 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 [SIP] 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.

6.2.2.1.3 Session Termination

To terminate a session, the OITF SHALL send a SIP BYE request to the IG that includes all headers listed in Table 8 and any other mandatory headers as per [SIP].

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP BYE includes all the mandatory headers as per [SIP] and the headers listed in Table 8. Following a successful validation, the IG SHALL generate a SIP BYE as specified in [TS124503] 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 [SIP].

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.

6.2.2.1.4 Session Refresh

It is the responsibility of the OITF to refresh the multicast content streaming session, as per [SES-TIMR] before the session expires. The IG SHALL consider a session terminated if it is not refreshed.

6.2.2.1.5 Content Reporting by an OIPF

To report watched content, the OITF SHALL send a SIP INFO request to the IG based one the SIP INFO framework, The SIP INFO request SHALL include all headers listed in Table 10 and all mandatory headers as per [SIP].

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INFO includes all mandatory headers as per [SIP] and the headers listed in Table 10. Following a successful validation, the IG SHALL generate a SIP INFO as specified in [TS124503] 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 [SIP].

6.2.2.1.6 Management of Content Reporting

At any time, the IG can receive a 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 section 6.2.1 SHALL validate that the SIP UPDATE includes all mandatory headers as per [SIP] 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. A SIP 200 OK response SHALL include the SIP headers defined in Table 13 and all mandatory headers as per [SIP].

6.2.2.1.7 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 a SIP re-INVITE to the IG that includes all the SIP headers in Table 5 and any other mandatory headers as per [SIP] (e.g. Via, Max-Forwards). The OITF SHALL also include an SDP offer conformant to step 1 in section 5.3.1.1.7.1, "User-initiated Activation of multicast content streaming Time Shift".

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INVITE includes all the mandatory headers as per [SIP] and the headers listed in Table 5. Following a successful validation, the IG SHALL generate an initial SIP INVITE request as specified in [TS124503] 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 [SIP].

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 section 5.3.1.1.7.1, "User-initiated Activation of multicast content streaming Time Shift." The OITF 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.

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 [SIP].

6.2.2.1.8 User-Initiated De-activation of Network-Based multicast content streaming Time Shift

To de-activate a time shift for a watched multicast content service, the OITF SHALL initiate a SIP re-INVITE to the IG that includes all the SIP headers in Table 5 and any other mandatory headers as per [SIP] (e.g. Via, Max-Forwards). The OITF SHALL also include an SDP offer conformant to step 1 in section 5.3.1.1.7.2, "User Initiated De-activation of multicast content steaming Time Shift".

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INVITE includes all the mandatory headers as per [SIP] and the headers listed in Table 5. Following a successful validation, the IG SHALL generate an initial SIP INVITE request as specified in [TS124503] 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 [SIP].

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 [SIP].

6.2.2.1.9 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 section 5.3.2.1.1 "Retrieval of Session Parameters" 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 section 5.3.1.1.1, "Session Initiation") 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.

6.2.2.2 Unicast content streaming with SIP session management

6.2.2.2.1 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 [SIP].

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP OPTIONS includes all the mandatory headers as per [SIP] and the headers listed in Table 14. Following a successful validation, the IG SHALL generate an initial SIP OPTIONS request as specified in [TS124503] 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 [SIP]. The SDP included in the SIP 200 OK response SHALL include the missing parameters according to section 6.1.2.2.1.2, “Protocol over NPI-4, NPI-19, NPI-26”.

6.2.2.2.2 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 [SIP] (e.g. Via, Max-Forwards). The OITF SHALL also include an SDP offer conformant to step 1 in section 5.3.2.1.2, “Session Initiation”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INVITE includes all the mandatory headers as per [SIP] and the headers listed in Table 16. Following a successful validation, the IG SHALL generate an initial SIP INVITE request as specified in [TS124503] 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 [SIP]. For a description of the received SDP answer, refer to section 5.3.2.1.2, “Session Initiation”.

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 [SIP].

6.2.2.2.3 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 [SIP].

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP BYE includes all the mandatory headers as per [SIP] and the headers listed in Table 19. Following a successful validation, the IG SHALL generate a SIP BYE as specified in [TS124503] 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 [SIP].

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.

6.2.2.2.4 Session Refresh

It is the responsibility of the OITF to refresh the unicast content streaming session, as per [SES-TIMR] before the session expires. The IG SHALL consider a session terminated if it is not refreshed.

6.2.2.3 Pay Per View multicast content service with SIP session management

6.2.2.3.1 PPV service initiation without existing multicast content streaming session

The OITF SHALL follow the same procedure for multicact content service in section 6.2.2.1.1, “Session Initiation” to initiate a PPV session. The SDP SHALL conform to sections 6.2.2.1.1, “Session Initiation” and 5.3.6.1.1, “PPV Service initiation without existing multicast content streaming session”.

6.2.2.3.2 Switching from a PPV service to a multicast content service

To join a multicact content service outside the set of negotiated channels for the ongoing scheduled session, the OITF SHALL request session modification as per section 6.2.2.1.2, “Session Modification.”

If the channel the OITF intends to joins is already negotiated, the OITF SHALL follow the normal procedures for leaving and joining multicast channels.

6.2.2.3.3 Switching to a PPV service from a multicact 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 section 6.2.2.1.2, “Session Modification.”

If the channel the OITF intends to joins is already negotiated, the OITF SHALL follow the normal procedures for leaving and joining multicast channels.

6.2.2.3.4 Session Termination

Session termination is identical to session termination of a multicast content service as depicted in section 6.1.2.1.1.3, “Session Termination.”

6.2.2.4 Parental Control for Scheduled Content using SIP

6.2.2.4.1 What is on TV – OITF initiated

6.2.2.4.1.1 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 [SIP] and that are missing from the table (e.g. Via, Max-Forwards).

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP SUBSCRIBE request includes all mandatory SIP headers as per [SIP] and the headers listed in Table 27. Following successful validation, the IG SHALL support the 3GPP IMS subscription to the event package as per [TS124503]. This includes inserting any additional mandatory SIP headers as REQUIRED by [TS124503].

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 section 6.2.1 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 [SIP].

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 [SIP].

The IG acting as a B2BUA in accordance with section 6.2.1 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 section 6.2.1 SHALL perform the same processing. The IG SHALL consider a subscription terminated if it is not renewed by the OITF.

6.2.2.4.1.2 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.

6.2.2.4.2 Parental Control

6.2.2.4.2.1 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 [SIP]. 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 section 6.2.1 SHALL validate that the SIP MESSAGE includes all the mandatory headers as per [SIP] and the headers listed in Table 31. Following a successful validation, the IG SHALL generate a SIP MESSAGE request to the network conformant to [SIP-IM] and [TS124503].

The IG acting as a B2BUA in accordance with section 6.2.1 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 [SIP].

6.2.2.4.2.2 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 section 6.2.1 SHALL validate that the incoming SIP MESSAGE is conformant to [SIP-IM] and [TS124503] and includes all the mandatory headers as per [SIP], [TS124503] 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 section 5.3.7.2.2, “Protocol for OITF Receiving a Request for Parental Control” 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 section 6.2.1 SHALL forward it to the network. The response SHALL include the SIP headers as per [SIP] and SHALL be validated by the IG before being sent to the network.

6.2.2.5 Network-based User Notification Services using SIP

6.2.2.5.1 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 [SIP] (e.g. Via, Max-Forwards). The body of the SIP MESSAGE SHALL include the MIME type defined in section 5.3.8.6, “XML Schema for Notification Request for Broadcast Reminder”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP MESSAGE includes all the mandatory headers as per [SIP] and the headers listed in Table 34. Following a successful validation, the IG SHALL generate a SIP MESSAGE request as specified in [TS124503] 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 [SIP].

6.2.2.6 Content Bookmarking using SIP

6.2.2.6.1 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 [SIP] (e.g. Via, Max-Forwards). The body of the SIP INFO message SHALL include the MIME type defined in section 5.3.9.5, “XML Schema for Content Bookmarking”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INFO includes all the mandatory headers as per [SIP] and the headers listed in Table 38. Following a successful validation, the IG SHALL generate a SIP INFO request as specified in [TS124503] 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 [SIP].

6.2.2.6.2 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 section 12 of [PSS-MBMS] from an OITF.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP UPDATE request includes all the mandatory headers as per [SIP] 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 [SIP].

6.2.2.6.3 Content-related bookmark retrieval

The OITF SHALL follow the procedure defined in section 6.2.2.2.2, “Session Initiation” for establishing a unicact 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 section 5.3.9.5, “XML Schema for Content Bookmarking” and related to the CoD-Bookmark Info Package.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INFO request includes all the mandatory headers as per [SIP] 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.

6.2.2.7 Local PVR Service using SIP

6.2.2.7.1 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 [SIP] (e.g. Via, Max-Forwards). The body of the SIP MESSAGE SHALL include the MIME type defined in section 5.3.10.2, “XML Schema for PVR Service”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP MESSAGE includes all the mandatory headers as per [SIP] and the headers listed in Table 42. Following a successful validation, the IG SHALL generate a SIP MESSAGE request as specified in [TS124503] 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 [SIP].

6.2.2.7.2 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 section 6.2.1 SHALL validate that the SIP MESSAGE request includes all the mandatory headers as per [SIP] and the headers listed in Table 44. The body of the SIP MESSAGE request SHALL include the XML schema defined in section 5.3.10.2, “XML Schema for PVR Service”. 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.

6.2.2.8 Network PVR (nPVR) using SIP

6.2.2.8.1 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 [SIP] (e.g. Via, Max-Forwards). The body of the SIP MESSAGE SHALL include the MIME type defined in section 5.3.10.2, “XML Schema for PVR Service” with the qualifications listed in section 5.3.11.1, “Protocol over HNI-IGI – HTTP Option”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP MESSAGE includes all the mandatory headers as per [SIP] and the headers listed in Table 42. Following a successful validation, the IG SHALL generate a SIP MESSAGE request as specified in [TS124503] 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 [SIP].

6.2.2.8.2 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 section 6.2.1 SHALL validate that the SIP MESSAGE request includes all the mandatory headers as per [SIP] and the headers listed in Table 44. The body of the SIP MESSAGE request SHALL include the XML schema defined in section 5.3.11.2, “XML schema for nPVR recording result”. 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.

6.2.2.9 Session Transfer using SIP

6.2.2.9.1 Generic Procedures

6.2.2.9.1.1 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 [SIP] 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 section 5.3.13.1.2.3, “Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode”.
- 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 section 5.3.13.1.2.3, “Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode”.

The OITF SHALL also include an SDP conformant to step 1 as defined in section 5.3.2.1.2, “Session Initiation”. The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the request includes all the mandatory SIP headers as per [SIP] and headers listed in Table 16. Furthermore, the IG perform the steps depicted in section 6.2.2.9.1.2, “IG handling of session transfers when the transferor and transferee are behind the same IG”.

Following that, the IG acting as a B2BUA in accordance with section 6.2.1 SHALL generate an initial SIP INVITE request to the network as specified in [TS124503] for originating sessions.

The IG acting as a B2BUA in accordance with section 6.2.1 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 [SIP]. For a description of the received SDP answer, refer to section 5.3.2.1.2, “Session Initiation”.

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 [SIP].

6.2.2.9.1.2 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 section 6.2.1 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 section 5.3.13.1.1.2, “IG handling of Session Initiation Requests related to a session transfer”. The SDP SHALL conform to step 1 in section 5.3.13.1.1.2, “IG handling of Session Initiation Requests related to a session transfer” as well.

The OITF SHALL conform to step 2 in section 5.3.13.1.1.2, “IG handling of Session Initiation Requests related to a session transfer” 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 section 5.3.13.1.1.2, “IG handling of Session Initiation Requests related to a session transfer”.

Subsequently, the transferor OITF SHALL receive a SIP ACK form the IG acting as a B2BUA in accordance with section 6.2.1 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 section 5.3.13.1.1.2, “IG handling of Session Initiation Requests related to a session transfer”.

Note that this entire procedure is bypassed if the transferor and transferee are not behind the same IG.

6.2.2.9.2 Session Transfer via Push Mode

6.2.2.9.2.1 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 section 6.2.3.2.4, “Procedure for Subscription to the Registration-State Event Package”.

Subsequently a target OITF (transferee OITF) can be selected from the returned information.

6.2.2.9.2.2 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 [SIP] (e.g. Via, Max-Forwards). The body of the SIP REFER message SHALL include the MIME type defined in section 5.3.13.2, “XML Schema for Session Transfer Information included in a session transfer request from the transferor to transferee”.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP REFER includes all the mandatory headers as per [SIP] and the headers listed in Table 46. Following a successful validation, the IG SHALL generate a SIP REFER request as specified in [TS124503] 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 [SIP].

Later at some point in time, the IG the SHALL receive a SIP NOTIFY request that reports the outcome of the session transfer request.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP NOTIFY request includes all the mandatory headers as per [SIP] 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.

6.2.2.9.2.3 Transferee OITF Handling of an Incoming Transfer

At some point in time, during a session transfer, the IG the SHALL receive a SIP REFER request destined for a transferee OITF.

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP REFER includes all the mandatory headers as per [SIP] and the headers listed in Table 46. The body of the SIP MESSAGE request SHALL include the XML schema defined in section 5.3.13.2, “XML Schema for Session Transfer Information included in a session transfer request from the transferor to transferee”. 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 section 5.3.13.1.2.3, “Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode”.

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 [SIP]. 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 section 5.3.13.1.2.3, “Transferee OITF Receiving an Incoming Session Transfer Request – Push Mode”, and proceed to initiate a new session according to the procedure in section 6.2.2.9.1.1, “Transferee Session Initiation Procedures Related to a Transferred Session”.

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 [SIP] (e.g. Via, Max-Forwards).

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP NOTIFY includes all the mandatory headers as per [SIP] and the headers listed in Table 48. Following a successful validation, the IG SHALL generate a SIP NOTIFY request as specified in [TS124503] 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 [SIP].

6.2.2.10 Purchase of Digital Media service using SIP

6.2.2.10.1 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 [SIP] (e.g. Via, Max-Forwards). The SIP INFO request body SHALL include an XML document as defined in section 5.3.5.8, “XML Schema for Purchase Request of Digital Media”, 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 [SIP] and the headers listed in Table 21. Following a successful validation, the IG SHALL generate a SIP INFO request as specified in [TS124503] 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 [SIP].

6.2.2.10.2 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 re-instate 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 [SIP] 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 [SIP].

6.2.2.11 Personalized Channel (Network Centric Model) using SIP

6.2.2.11.1 Request Initiation for Personal Channel

To initiate a session for a personalized channel, the OITF SHALL follow section 6.2.2.2.2, “Session Initiation” 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 section 6.2.1 SHALL follows as well the same steps in section 6.2.2.2.2.

6.2.3 Protocols for Service Access and Control Functions

6.2.3.1 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 [SIP] 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 section 6.2.1 SHALL validate that the SIP SUBSCRIBE request includes all the mandatory headers as per [SIP] and the headers listed in Table 50. Following a successful validation, the IG SHALL follow the procedure in section 6.1.3.1, “Service Provider Discovery”, when it comes to initiating a SIP SUBSCRIBE towards the Service Provider.

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL return to the OITF the response received from the network as per step 3b in section 6.1.3.1, “Service Provider Discovery”.

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.

6.2.3.2 User Registration and Network Authentication

6.2.3.2.1 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 58. In addition, the OITF SHALL include all mandatory SIP headers as per [SIP] and that are missing from the table (e.g., Via, Max-Forwards).

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP REGISTER request includes all the mandatory headers as per [SIP] and the headers listed in Table 58. Following a successful validation, the IG SHALL support the 3GPP IMS registration procedure as per TS 124 503 [TS124503]. This includes handling of user authentication and authorization., and including any additional SIP headers that are mandatory based on TS 124 503 [TS124503].

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 59.

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.

6.2.3.2.2 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 58. 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 [SIP] and that are missing from the table (e.g., Via, Max-Forwards).

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP REGISTER request includes all the mandatory headers as per [SIP] and the headers listed in Table 58. Following a successful validation, the IG SHALL support the 3GPP IMS de-registration procedure as per TS 124 503 [TS124503]. 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 inTable 59.

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).

6.2.3.2.3 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.

6.2.3.2.4 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 60. In addition, the OITF SHALL include all mandatory SIP headers as per [SIP] and that are missing from the table (e.g., Via, Max-Forwards).

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP SUBSCRIBE request includes all the mandatory headers as per [SIP] and the headers listed in Table 60. Following a successful validation, the IG SHALL support the 3GPP IMS subscription to registration-event package as per TS 124 503 [TS124503]. This includes inserting any additional SIP headers that are mandatory based on TS 124 503 [TS124503].

The IG acting as a B2BUA in accordance with section 6.2.1 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 section 6.2.1 SHALL validate the incoming NOTIFY request before sending it to the OITF. The incoming NOTIFY request SHALL include the SIP headers depicted in Table 62 and any other mandatory headers as per [SIP].

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 63. The IG acting as a B2BUA in accordance with section 6.2.1 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 section 6.2.1 SHALL perform the same processing.

6.2.3.2.5 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.

6.2.3.3 Notification of Service Profile changes

6.2.3.3.1 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 [SIP] 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 section 6.2.1 SHALL validate that the SIP SUBSCRIBE request includes all the mandatory headers as per [SIP] 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 [TS124503], [SIP-EVNT] and [XCAP-EVT].

The IG acting as a B2BUA in accordance with section 6.2.1 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 section 6.2.1 SHALL validate the incoming NOTIFY request includes the SIP headers depicted in Table 56 and any other mandatory headers as per [SIP].

The OITF SHALL validate that the SIP body is compliant to the XML schema associated with application/XCAP-diff+xml as defined in [XCAP-EVT] and [XCAP-DFF]. 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 section 6.2.1 SHALL first validate the SIP 200 OK response (or any other received response) before sending it to the network.

For all subsequent SIP NOTIOFY requests received using the same SIP dialog, the IG acting as a B2BUA in accordance with section 6.2.1 SHALL perform the same processing. The IG SHALL consider a subscription terminated if it is not renewed by the OITF.

6.2.3.3.2 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.

6.2.4 Protocols for Communication Services using SIP

6.2.4.1 Procedure for Instant Message Based CallerID

The procedure for caller id is identical to the reception of an instant message as depicted in section 6.2.4.2.2, “Procedure for Incoming Instant Messaging”.

6.2.4.2 Instant Messaging

6.2.4.2.1 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 69 and any other mandatory headers as per [SIP]. The content of the SIP MESSAGE SHALL be conformant to RFC 3428 [SIP-IM].

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP MESSAGE includes all the mandatory headers as per [SIP] and the headers listed in Table 69. Following a successful validation, the IG SHALL generate a SIP MESSAGE request to the network conformant to [SMPL-PRES].

The IG SHALL forward any received SIP response to the OITF. A SIP 200 OK response SHALL include the SIP headers defined in Table 70 and all mandatory headers as per [SIP].

6.2.4.2.2 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 section 6.2.1 SHALL validate that the incoming SIP MESSAGE is conformant to [SMPL-PRES] and includes all the mandatory headers as per [SIP] and all the headers listed in Table 71. 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 section 6.2.1 SHALL forward it to the network. The response SHALL include the SIP headers in Table 70 and SHALL be validated by the IG before being sent to the network.

6.2.4.3 Presence

6.2.4.3.1 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 88. In addition, the OITF SHALL include all mandatory SIP headers as per [SIP] and that are missing from the table (e.g. Via, Max-Forwards).

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP SUBSCRIBE request includes all the mandatory headers as per [SIP] and the headers listed in Table 88. Following successful validation, the IG SHALL support the 3GPP IMS subscription to presence event package as per [SMPL-PRES]. 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 [SMPL-PRES] before sending it to the OITF. The incoming NOTIFY request SHALL include the SIP headers depicted in Table 90 and any other mandatory headers as per [SIP].

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 91. 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 section 6.2.1 SHALL perform the same processing. The IG SHALL consider a subscription terminated if it is not renewed by the OITF.

6.2.4.3.2 Procedure for Terminating a Subscription to Presence Information

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

6.2.4.3.3 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 92 and other mandatory headers as per [SIP]. The content of the SIP PUBLSIH SHALL be based on section 5.5.4.6, “Presence Notification and Publish Schema”.

The IG acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP PUBLISH request includes all the mandatory headers as per [SIP] and the headers listed in Table 92. Following a successful validation, the IG SHALL generate a SIP PUBLISH request to the network conformant to [SMPL-PRES].

The IG SHALL forward any received SIP response to the OITF. A SIP 200 OK response SHALL include the SIP headers defined in Table 93 and all mandatory headers as per [SIP].

6.2.4.4 Chatting

For IM sessions, the OITF SHALL implement the MSRP stack conformant to [SMPL-IM].

6.2.4.4.1 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 72. 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 [SIP] (e.g. Via, Max-Forwards).

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INVITE request includes all the mandatory headers as per [SIP] and the headers listed in Table 72. Following a successful validation, the IG SHALL generate an initial INVITE request as specified in [SMPL-IM].

The IG acting as a B2BUA in accordance with section 6.2.1 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 73 and all mandatory headers as per [SIP].

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 74 and all mandatory headers as per [SIP].

6.2.4.4.2 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 section 6.2.1 SHALL validate that the incoming SIP INVITE is conformant to [SMPL-IM] and includes all the mandatory headers as per [SIP] and all the headers listed in Table 85. 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 [RFC4975]. 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 section 6.2.1 SHALL forward it to the network. The response SHALL include the SIP headers in Table 86 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 87 and all mandatory headers as per [SIP].

6.2.4.4.3 Session Termination

To terminate a session, the OITF SHALL send a SIP BYE request to the IG that includes all headers listed in Table 81 and any other mandatory headers as per [SIP].

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP BYE request includes all the mandatory headers as per [SIP] and the headers listed in Table 81. Following a successful validation, the IG SHALL generate a SIP BYE request as specified in [SMPL-IM].The IG SHALL forward the received response to the OITF. A SIP 200 OK response SHALL include the SIP headers defined in Table 82 and all mandatory headers as per [SIP].

Alternatively, on receipt of a SIP BYE request from the IPTV Control FE, the IG acting as a B2BUA in accordance with section 6.2.1, after validating the request, to include all headers listed in Table 83 and any other mandatory headers as per [SIP], 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.

6.2.4.4.4 Session Refresh

It is the responsibility of the OITF to refresh the IM session, as per [SES-TIMR] before the session expires. The IG SHALL consider a session terminated if it is not refreshed, and SHALL clear its internal state accordingly.

6.2.4.4.5 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 [RFC4975] and [SMPL-IM].

6.2.4.5 Content Sharing

6.2.4.5.1 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 INVIVE to the IG that includes all the SIP headers and any other mandatory headers as per [SIP]. SDP shall be used as specified in [TS124503].

The IG SHALL validate that the SIP INVITE includes all the mandatory headers as per [SIP]. Following a successful validation, the IG SHALL generate a SIP INVITE request to the network conformant to [SIP].

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 [SIP].

- The IG SHALL forward any incoming SIP REFER, for the purpose of session transfer, that conforms to the [TS24237] 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 [RFC3515].
- The IG SHALL forward any incoming SIP NOTIFY, reporting the outcome of a session transfer associated with a REFER request, that conforms to [RFC3515] 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).

6.2.4.5.2 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 [SMPL-IM] and includes all the mandatory headers as per [SIP]. 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 [TS124503]. 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.

6.2.4.5.3 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 [SIP].

The IG, acting as a B2B UA SHALL validate that the SIP BYE request includes all the mandatory headers as per [SIP]. Following a successful validation, the IG SHALL generate a SIP BYE request as specified in [SIP].The IG SHALL forward the received response to the OITF. A SIP 200 OK response SHALL include all the mandatory headers as per [SIP].

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 [SIP], 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.

6.2.4.6 Multimedia (MM) Telephony

6.2.4.6.1 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 132. The body of the SIP INVITE SHALL be populated in accordance with [SDP] reflecting the appropriate description for each media, and corresponding codecs for MM session. Section 6.2.4.6.8 entitled “SDP Media Type parameters for media description” details the SDP representation for codecs that can be used by the OITF.

The OITF SHALL also include any other mandatory headers as per [SIP] (e.g. Via, Max-Forwards).

The IG, being service aware, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP INVITE request includes all the mandatory headers as per [SIP] and the headers listed in Table 132. 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 [TS124503] and [TS181005] and SHALL include all mandatory headers.

The IG acting as a B2BUA in accordance with section 6.2.1 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 133 and all mandatory headers as per [SIP].

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 134 and all mandatory headers as per [SIP]. The IG acting as a B2BUA, in accordance with section 6.2.1, SHALL forward the SIP ACK to the ASM.

6.2.4.6.2 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 [TS124503] and [TS181005] and includes all the mandatory headers as per [SIP] and all the headers listed in Table 135. Following the successful validation, the IG acting as a B2BUA in accordance with section 6.2.1 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 [TS124503] and [TS181005] and the mandatory information pertinent to the chosen codecs in accordance with section 6.2.4.6.8 entitled “SDP Media Type parameters for media description”. 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 [TS124503] and [TS181005].

When the SIP response is received from the OITF, the IG acting as a B2BUA in accordance with section 6.2.1, SHALL forward it to the network. The response SHALL include the SIP headers in Table 136 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 section 6.2.1 SHALL forward the received SIP ACK from the network to the OITF. The SIP ACK SHALL include the SIP headers defined in Table 137 and all mandatory headers as per [SIP].

6.2.4.6.3 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 138 and any other mandatory headers as per [SIP].

The IG, acting as a B2BUA in accordance with section 6.2.1 SHALL validate that the SIP BYE request includes all the mandatory headers as per [SIP] and the headers listed in Table 138. Following a successful validation, the IG SHALL generate a SIP BYE request as specified in [TS124503] and [TS181005] towards the ASM. The IG acting as a B2BUA in accordance with section 6.2.1 SHALL forward the received response to the OITF. A SIP 200 OK response SHALL include the SIP headers defined in Table 139 and all mandatory headers as per [SIP].

6.2.4.6.4 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 140 and any other mandatory headers as per [SIP], and acting as a B2BUA in accordance with section 6.2.1 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 141, and which the IG, acting as a B2BUA in accordance with section 6.2.1 SHALL forward to the network.

6.2.4.6.5 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.

6.2.4.6.6 Session Refresh

It is the responsibility of the OITF to refresh the MM session, as per [SES-TIMR] before the session expires. The IG SHALL consider a session terminated if it is not refreshed, and SHALL clear its internal state accordingly.

6.2.4.6.7 Tables of Message elements

Table 132: 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	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
From	RFC 3261 [SIP]
To The URI part of To SHALL be set to the value of the Request URI in the “Request-Line”	RFC 3261 [SIP]
Contact If GRUU has not been requested at registration, then the URI parameter and the sip.instance feature tag MUST be included and MUST 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	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Content-Type	RFC 3261 [SIP] (application/sdp)
Content-Length	RFC 3261 [SIP]
Supported If GRUU is used, Supported SHALL also be set to “gruu”	RFC 3261 [SIP] set to timer
Session-Expires	RFC 4028 [SES-TIMR]

Table 133: 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	RFC 3261 [SIP] SIP/2.0 <response>
From	RFC 3261 [SIP]

To	RFC 3261 [SIP]
Contact	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Session-Expires	RFC 4028 [SES-TIMR]
Content-Type	RFC 3261 [SIP]
Content-Length	RFC 3261 [SIP]

Table 134: 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	RFC 3261 [SIP] ACK <Request URI> SIP/2.0
From	RFC 3261 [SIP]
To The URI part of To SHALL be set to the value of the Request URI in the “Request-Line” of the initial request	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Contact The URI parameter MUST be included, and MUST match what has been inserted in the INVITE message. The IG includes all other mandatory parameters that are absent.	RFC 3261 [SIP]

Table 135: 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)	RFC 3261 [SIP] INVITE <Request URI> SIP/2.0
From	RFC 3261 [SIP]
To The URI part of To SHALL be set to the value of the	RFC 3261 [SIP]

Request URI in the “Request-Line”	
Contact	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Content-Type	RFC 3261 [SIP] (application/sdp)
Content-Length	RFC 3261 [SIP]
Supported	RFC 3261 [SIP] set to timer, and/or GRUU if supported by remote user
Session-Expires	RFC 4028 [SES-TIMR]

Table 136: 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	RFC 3261 [SIP] SIP/2.0 <response>
From	RFC 3261 [SIP]
To	RFC 3261 [SIP]
Contact If GRUU has not been requested at registration, then the URI parameter and the sip.instance feature tag MUST be included and MUST 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.	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Session-Expires	RFC 4028 [SES-TIMR]
Content-Type	RFC 3261 [SIP]
Content-Length	RFC 3261 [SIP]

Table 137: 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	RFC 3261 [SIP] ACK <Request URI> SIP/2.0
From	RFC 3261 [SIP]
To The URI part of To SHALL be set to the value of the Request URI in the “Request-Line” of the initial request	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Contact The URI parameter MUST be included, and MUST match what has been included in the initial INVITE message..	RFC 3261 [SIP]

Table 138: List of SIP headers for an Outgoing SIP BYE (OITF→IG)

SIP Header	Source of Coding Information
Request-Line Note: The Request URI MUST match the contact URI included in the contact field of the SIP INVITE (for outgoing session) or a 200 OK (for incoming session)	RFC 3261 [SIP] BYE <Request URI> SIP/ 2.0
From	RFC 3261 [SIP]
To	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Content-Length MUST be set to 0	RFC 3261 [SIP]
Contact	RFC 3261 [SIP]

Table 139: List of SIP headers for the response to an Outgoing SIP BYE (IG→OITF)

SIP Header	Source of Coding Information
Response-Line	RFC 3261 [SIP] SIP/2.0 <response>
From	RFC 3261 [SIP]

To	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]

Table 140: List of SIP headers for an Incoming SIP BYE (IG→OITF)

SIP Header	Source of Coding Information
Request-Line Note: The Request URI MUST match the contact URI included in the contact field of the SIP INVITE (for outgoing session) or a 200 OK (for incoming session)	RFC 3261 [SIP] BYE <Request URI> SIP/ 2.0
From	RFC 3261 [SIP]
To	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]
Content-Length MUST be set to 0	RFC 3261 [SIP]
Contact	RFC 3261 [SIP]

Table 141: List of SIP headers for the response to an Incoming SIP BYE (OITF→IG)

SIP Header	Source of Coding Information
Response-Line	RFC 3261 [SIP] SIP/2.0 <response>
From	RFC 3261 [SIP]
To	RFC 3261 [SIP]
Call-ID	RFC 3261 [SIP]
CSeq	RFC 3261 [SIP]

6.2.4.6.8 SDP Media Type parameters for media description

This section describes the media configuration in SDP for all audio and video codecs defined in [OIPF_MEDIA2] for multimedia telephony services.

- Media type parameters to configure H.264 video RTP payload types SHALL comply with [RFC3984].
- Media type parameters to configure MPEG-4 Part-2 video RTP payload types SHALL comply with [RFC3016].
- Media type parameters to configure H.263 video RTP payload types SHALL comply with [RFC4627].

- Media type parameters to configure G.711 or G.722 audio RTP payload types SHALL comply with [RFC3551].
- Media type parameters to configure AMR-NB or AMR-WB RTP payload types SHALL comply with [RFC4867].
- Media type parameters to configure G.719 payload types SHALL comply with [RFC5404].
- Media type parameters to configure MPEG-4 AAC-LD/-ELD payload types SHALL comply with [RFC3640].

Note: RFC 3640 [RFC3640] 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 [AAC]. The AudioSpecificConfig() contains the audioObjectType, that shall be set to “39” for AACELD and “23” for AACLD.

Example SDP for AACLD, stereo 48kHz:

```
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 SDP for G.719, stereo 48kHz:

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96  G719/48000/2
```

7 RTSP

This section defines the protocol for the use of RTSP over the following reference points:

- UNIS-11
- NPI-10

7.1 Protocols for IPTV Service Functions

7.1.1 Use of RTSP for unicast content streaming services

7.1.1.1 RTSP Profile without SIP session management over UNIS-11 and NPI-10

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 RFC 2326 [RTSP] and with the following profile.

The following describes the RTSP Profile for this Release. The functionalities not identified in this section are out of scope of OIPF:

- NPT Range time format SHALL be supported by clients and servers, section 3.6 of [RTSP].
- The extension mechanism using option tags in section 3.8 of [RTSP] 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 (section 1.4 of [RTSP])
    - 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 section 9.1 of [RTSP], “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 section 10.12, “Embedded (Interleaved) Binary Data” of [RTSP].

Clients SHALL NOT use PLAY messages in the PLAY state as keep-alives (section 10.5 of [RTSP]).

Servers SHALL NOT use RTSP over UDP, see Appendix G of [RTSP] and related functionality like the rtspu URI scheme in section 22.14.3 of [RTSP].

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 [RFC3986], which extends the encodings and conventions that were hinted at but not specified in [RTSP].
- Regarding section 9, “Connections” of [RTSP], 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 Annex F.6.2, “NAT Traversal and keep-alive messages for unicast content streaming services”) 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 section 12.17, “CSeq” of [RTSP], 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 MUST 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 [RTSP], these are indicated by using open-ended time intervals via a=range attribute in SDP. E.g., 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 sections 4.3 and 12.14 of [RTSP], because section 4.3 of [RTSP] refers to HTTP/1.1 (which only recommends it) while section 12.14 of [RTSP] clearly says that it MUST be included.

Regarding RTP-Info, section 12.33 of [RTSP]:

- 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 RFC 3550 [RTP].
- 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.

7.1.1.1.1 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 RFC 4566 [SDP] 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.

7.1.1.1.2 RTSP Control for media delivery

7.1.1.1.2.1 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.

7.1.1.1.2.2 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_PAREMETER 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 seconds 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.

7.1.1.1.2.3 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 [RTSP2-AN].

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, etc.). The OITF SHALL respond with a RTSP 200 OK.

7.1.1.1.2.4 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.

7.1.1.1.2.5 Handling of CDF-initiated session termination

If the CFD 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 [RTSP2-AN].

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 section 7.1.1.1.3, “RTSP Session Teardown”.

7.1.1.1.2.6 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 [RTSP2-AN].

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 section 7.1.1.1.3, “RTSP Session Teardown”.

7.1.1.1.3 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.

7.1.1.1.4 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.

7.1.1.2 RTSP profile with SIP session management over UNIS -11 and NPI-10

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 section 7.2.1, “Performance Monitoring over UNIT-18.”).
- The a=rtcp-xr line includes information on the sample performance metrics the Service Provider requests from the client for that session (see section 9.2.1, “Performance Monitoring over UNIT-18.”)
- Finally, the b=RR line specifies the reporting bandwidth assigned to the OITF in bits per second, see section 2 of RFC 3556 [SDP-RTCP]. 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 RFC 2326 [RTSP] with modifications defined by this specification.

Guidelines for specifying cumulative metrics to be conveyed using the RTSP Headers defined in [OIPF_PROTEX2]. Similarly to the RTCP case, “OIPF-BasicPerfMonCumulSubset1” is used in this specification as a placeholder and for illustrative purposes.

7.1.1.2.1 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.

7.1.1.2.2 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 section 6.1.2.1.3, “Content Reporting and Management of Content Reporting.”)

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.

7.1.1.2.3 RTSP Control for media delivery

7.1.1.2.3.1 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 [TS183063] section 7.1.1.2 with the OITF acting as a UE and the Cluster Controller acting as a MCF.

7.1.1.2.3.2 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 [TS183063] section 7.1.1.3 with the OITF acting as a UE and the Cluster Controller acting as a 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.

7.1.1.2.3.3 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.

7.1.1.2.3.4 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 [RTSP2-AN]. Note that the RTSP version used in this specification is “1.0” and not “2.0” as in the examples in [RTSP2-AN].

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.

7.1.1.2.3.5 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 perform the following:

- Return an RTSP 200 OK to the CDF,
- Proceed to tear down the RTSP session in accordance with section 7.1.1.1.3, “RTSP Session Teardown” and where the cluster controller plays the role of the OITF,
- Proceed to tear down the IMS session with the CDNCF.

7.1.1.2.3.6 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 perform the following:

- Proceed to tear down the RTSP session in accordance with section 7.1.1.1.3, “RTSP Session Teardown” and where the cluster controller plays the role of the OITF,
- Proceed to tear down the IMS session with the CDNCF.

7.1.1.2.4 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).

7.1.1.2.5 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.

7.1.2 Use of RTSP for Forced Play Out Control

7.1.2.1 RTSP for managed model over UNIS-11 and NPI-10

The RTSP/RTCP monitoring, RTSP session setup and teardown procedures are unchanged by Forced Play Out and SHALL follow the description in section 7.1.1, “Use of RTSP for unicast content streaming services”.

7.1.2.1.1 RTSP Control for media delivery

7.1.2.1.1.1 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 section 7.1.1.2.3, “RTSP Control for media delivery.”

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 section 7.1.1.2.3, “RTSP Control for media delivery.”

Handling of End of Stream is not affected by Forced Play Out and is the same as specified in section 7.1.1.2.3, “RTSP Control for media delivery.”

7.1.3 Use of RTSP for Network PVR (nPVR)

7.1.3.1 RTSP Session Setup

When performing recording, the Cluster Controller SHALL first retrieve from the incoming request the SDP of the scheduled content program 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 RFC 4566 [SDP]. The Cluster Controller SHALL then issue a SETUP request to CDF. The CDF then SHALL join the multicast group as defined in section 8.1.3.1, “Protocol over NPI-40”.

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 start recording the content. When receiving error messages from the CDF, the Cluster Controller SHALL try another CDF.

7.1.3.2 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.

7.1.3.3 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.

7.1.4 Use of RTSP for Personalized Channel

7.1.4.1 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 RFC 3376 [IGMP3].

7.1.4.2 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 indicae 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 RFC 3376 [IGMP3].

7.1.5 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 sections 7.1.1.2.3, "RTSP Control for media delivery" and 7.1.1.2.4, "RTSP Control for media delivery" 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 section 5.3.9.4.1, "Protocol for Retrieving Stored Content Bookmarks" to retrieve the content bookmark, and subsequently it can use the procedure defined in Step 1.

7.2 Protocols for Service Access and Control

7.2.1 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 section 9.2.1, "Performance Monitoring over UNIT-18"), 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 section 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 section 7.1.1.2.2, "RTSP Session Setup."

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 MUST 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 section 5.3.3.6 of TS26.234v750 [PSS].

The ABNF definition (See RFC 4234 [ABNF]) 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 RFC 2326 [RTSP]

    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 [PSS], 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 [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 RFC 2326 [RTSP]

     Rtsp-URL = as defined in RFC 2326 [RTSP]

     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.

ABNF [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 RFC 2326 [RTSP]

     Measure-Range =  “range” “:” Ranges-Specifier

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

     Measure = Value [SP Timestamp]

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

     Timestamp = NPT-Time

     NPT-Time = as defined in RFC 2326 [RTSP]

     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 specification. 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 [PSS]. The semantics SHALL comply with section 5.3.2.3 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.

8 IGMP and Multicast Protocol

This section 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

8.1 Protocols for IPTV Service Functions

8.1.1 Multicast content streaming service on UNIS-13

The OITF SHALL support IGMPv3 as described in RFC 3376 [IGMP3]. 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 [IGMP3] section 7.

8.1.1.1 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 [TS183063] sections 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.

8.1.1.2 Procedure for multicast content streaming

The use of IGMP on UNIS-13 without session initiation SHALL be as defined in section 8.1.1.1, “Procedure for multicast content streaming with SIP session management”.

In the case there is no session initiation, the procedures described in section 8.1.2 of [TS183063] to set the Group/Multicast Address Records will not retrieve any channel or source address information from the session initiation step.

8.1.1.3 Procedure for multicast RET

The use of IGMP on UNIS-13 SHALL be defined as in section 8.1.1.1, “Procedure for multicast content streaming with SIP session management” or as defined in section 8.1.1.2, “Procedure for multicast content streaming”.

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 TS 102 034 [TS102034] and Annex M.1.3, “Multicast RET for multicast content service” in this specification.

8.1.2 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 section 8.1.1, “Multicast content streaming service on UNIS-13.”

8.1.3 nPVR

8.1.3.1 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 section 8.1.1.1, “Procedure for multicast content streaming with SIP session management”. 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.

8.1.4 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 section 9.1.3, “nPVR”.

For the OITF, Network Generated Notification session initiation over UNIS-13 SHALL be the same as multicast content streaming session initiation as described in section 8.1.1, “Multicast content streaming service on UNIS-13”.

8.1.5 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 section 9.1.3, “nPVR”.

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 [OIPF_META2].

8.2 Protocols for Service Access and Control Functions

8.2.1 Service Discovery and Content Selection

8.2.1.1 Protocol over UNIS-15 and UNIS-7

DVB SD&S Transport Protocol (DVBSTP) specified in section 5.4.1 of [TS102034] SHALL be used to transport the Service Discovery and Content Metadata related information over multicast.

8.2.1.2 Protocol over UNIS-13

The use of IGMP on UNIS-13 for Service Discovery and Content Selection SHALL be as defined in [TS183063] section 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.

8.2.2 Remote Management

8.2.2.1 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 [TR069].

8.3 Protocols for System Infrastructure Functions

8.3.1 Interactive application delivery

8.3.1.1 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 [RFC3926] SHALL be used when DAE and PAE applications are delivered through multicast. The FDT-Instance XML Schema defined in [RFC3926] 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.

9 RTP/RTCP

This section defines the protocol for the use of RTP and RTCP over the following reference points:

- UNIT-17
- UNIT-18
- NPI-41

9.1 Protocols for IPTV Service Functions

9.1.1 Multicast content streaming service

Scheduled content may be delivered as a multicast stream over UNIT-17.

9.1.1.1 Protocol over UNIT-17

The use of RTP on this reference point SHALL comply with [TS102034] section 7.1 and subsection 7.1.1.

9.1.2 unicast content streamin service

Streamed Scheduled Content or Content on Demand may bedelivered as a unicast stream over UNIT-17. RTP and HTTP MAY be used. This section specifies the use of RTP.

9.1.2.1 Protocol over UNIT-17

RTP SHALL be used on this reference point and SHALL comply with [TS102034] section 7.1 and subsection 7.1.1.

The use of RTCP SHALL be compliant to section 9.2.1.

9.1.3 nPVR

9.1.3.1 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 [TS102034] section 7.1 and subsection 7.1.1.

9.2 Service Access and Control

9.2.1 Performance Monitoring over UNIT-18

This section 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 subsection 7.1.1.1 of [TS102034] with the following additions:

- RTCP Receiver Reports defined by RFC 3550 [RTP] SHALL be used to include RTCP XR extended reports following the rules of RFC 3611 [RTCP-XR] 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 RFC 3556 [SDP-RTCP]. See section 7.1.1.2 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 RFC 3611 [RTCP-XR]. 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 section 5.1 of RFC 3611 [RTCP-XR] are as follows:

```
rtcp-xr-attrib = “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 [OIPF_PROTEX2]. *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 RFC 3611 [RTCP-XR] (e.g., pkt-loss-rle, pkt-dup-rle, pkt-rcpt-times, etc...) but these are not REQUIRED for the performance monitoring to work correctly.

9.3 Application Layer Forward Error Correction

This section 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 is conforming to [TS102034] Annex E. 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 [TS102034] section 5.2.6.2.

9.4 Application Layer Retransmission (RET)

This section 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 [TS102034].

9.4.1 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 [TS102034].

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 [TS102034].

9.4.2 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 [TS102034].

9.5 Fast Channel Change (FCC)

This section 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 [FCC].

9.5.1 Protocol over UNIT-17

The unicast Fast Channel Change RTP burst data is transferred over the UNIT-17, in conformance with [FCC].

9.5.2 Protocol over UNIT-18

RTCP SHALL be used on this reference point and SHALL comply with [FCC].

9.6 Protocols for Multimedia Telephony Services

9.6.1 Protocol over UNIT-17

The use of RTP on this reference point SHALL comply with [RFC3550] and [RFC3551].

9.6.1.1 RTP payload formats for video streams

The RTP payload format defined in [RFC3984] SHALL be used for the transport of H.264 video streams according to the video media type AVC_VDC as defined in [OIPF_MEDIA2]. The following restrictions on the usage of [RFC3984] 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 [RFC3016] SHALL be used for the transport of MPEG-4 Part-2 video streams according to the video media type MP4V as defined in [OIPF_MEDIA2].

The RTP payload format defined in [RFC4629] SHALL be used for the transport of H.263 video streams according to the video media type H263 as defined in [OIPF_MEDIA2].

9.6.1.2 RTP payload formats for audio streams

The RTP payload formats defined in [RFC3551] 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 [OIPF_MEDIA2].

The RTP payload format defined in [RFC4867] and further specified in chapter 7.4.2 of [TS26.114] 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 [OIPF_MEDIA2].

The RTP payload format defined in [RFC4749] SHALL be used for the transport of G.729.1 speech audio streams according to the audio media type G7291 as defined in [OIPF_MEDIA2].

The RTP payload format defined in [RFC5404] SHALL be used for the transport of G.719 audio streams according to the audio media type G719 as defined in [OIPF_MEDIA2].

The RTP payload format defined in [RFC3640] SHALL be used for the transport of MPEG-4 audio streams according to the audio media types AACLD and AACELD as defined in [OIPF_MEDIA2].

10 UPnP

10.1 Protocols for System Infrastructure Functions

10.1.1 UPnP Discovery

OITFs that do not support native HNI-IGI do not support UPnP-based IG discovery.

10.1.1.1 Procedure for IG Discovery

10.1.1.1.1 Discovery Sequence

When an OITF powers up, the OITF SHALL automatically discover the IG using the UPnP Discovery Mechanism defined by UPnP Device Architecture [UPNP]. 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.

10.1.1.1.2 urn:oipf-org:device:ig:1 device definitions

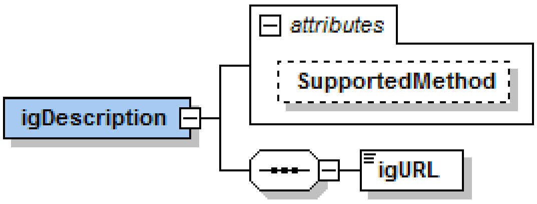
This section defines the urn:oipf-org:device:ig:1 deviceType.

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.

10.1.1.1.3 IG Description

To interact with the IG, the OITF MUST 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 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>
```

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 (section 5.6.1, “OITF-IG Interface (HNI-IGI)”), 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 section 5.6.1, “OITF-IG Interface (HNI-IGI).”)

	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 [OIPF_CSP2] section 5.4.5 or the, more general, procedure, HTTP Digest Authentication using IMS Gateway in [OIPF_CSP2] section 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>
```

10.1.1.2 Procedure for AG Discovery

10.1.1.2.1 Discovery Sequence

When an OITF powers up, the OITF SHALL automatically discover the AG using the UPnP Discovery Mechanism defined by UPnP Device Architecture [UPNP]. 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.

10.1.1.2.2 urn:oipf-org:device:ag:1 device definitions

This section defines the 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.

10.1.1.2.3 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 [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 MUST 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:

```
<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>
```

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”.

10.1.1.3 Procedure for CSPG-DTCP Discovery

10.1.1.3.1 Discovery Sequence

When an OITF powers up, the OITF SHALL automatically discover the CSPG-DTCP using the UPnP Discovery Mechanism defined by UPnP Device Architecture [UPNP]. 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 CGPG-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.

10.1.1.3.2 urn:oipf-org:device:cspg-dtcp:1 device definitions

This section defines the 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.

10.1.1.3.3 CSPG-DTCP Description

To interact with the CSPG-DTCP, the OITF MUST 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: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>
```

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 [OIPF_META2].

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>
```

10.1.1.4 Procedure for OITF Discovery

10.1.1.4.1 Discovery Sequence

When an OITF powers up, the IG SHALL automatically discover the OITF using the UPnP Discovery Mechanism defined by UPnP Device Architecture [UPNP]. 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:oitf: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.

10.1.1.4.2 OITF Description

This section defines the urn:oipf-org:device:oitf:1 deviceType.

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 ([UPNP-MR], [UPNP-MS]). Detailed descriptions and schemas of the ServiceTypes are defined in these specifications.

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.

11 DLNA

11.1 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 [DLNA].

12 DHCP

This section defines the protocol for the use of DHCP over the following reference points:

- UNIT-16

12.1 Protocols for System Infrastructure Functions

12.1.1 Network Attachment

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: 2 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.

12.1.1.1 DHCP Option Usage

12.1.1.1.1 Common Options

The following is the minimum set of DHCP options defined in RFC 3442 [CLSLESS] and RFC 2132 [DHCP-OPT] 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 [SHA-1]. The domain name SHALL be set to the domain name received via DHCP option 15 (see section 12.1.1.1.2, "Option 15.")

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 section 7.1.1 of TS 183 019 [TS183019].

12.1.1.1.2 Option 15

Option 15 SHALL be used by the OITF to request the domain name used to generate the device identity (deviceID) as per section 6.1.3.2.1, “User Identity Modelling”.

12.1.1.1.3 Option 124/125

The OITF SHALL send a Vendor–Identifying Vendor Class option 124 as specified in RFC 3925 [DHCP-VND] 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 RFC 3925 [DHCP-VND], 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 Annex F.1, “OITF Start up High-Level Procedure”, or
- The FQDN/IP address of the Service Provider Discovery Functional Entity (SP Discovery FE), as per section 6.1.3.1, “Service Provider Discovery”.

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 [FCC], 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: for this release it is not defined how to configure the DHCP server on the WAN gateway for FCC/RET server location.

13 UDP

13.1 Protocols for IPTV Service Functions

13.1.1 Multicast content streaming service

13.1.1.1 Protocol over UNIT-17

When MPEG2-TS are encapsulated directly in UDP (User Datagram Protocol), the encapsulation SHALL conform to TS 102 034 [TS102034] section 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 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 then RTP encapsulation is used).

13.1.2 Unicast content streaming service

13.1.2.1 Protocol over UNIT-17

The use of UDP on this reference point SHALL be as specified in section 13.1.1.1, “Protocol over UNIT-17.”

13.1.3 nPVR

13.1.3.1 Protocol over NPI-41

The use of UDP on this reference point SHALL be as specified in section 13.1.1.1, “Protocol over UNIT-17.”

14 FLUTE

This section defines the protocol for the use of FLUTE over the following reference points:

- UNIT-19

14.1 Protocols for IPTV Service Functions

14.1.1 Emergency Notification

Emergency notification is delivered in a multicast session over UNIT-19.

14.1.1.1 Protocol over UNIT-19

The use of FLUTE SHALL comply with [RFC3926]. 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 [RFC3926] with the extension defined in section 8.3.1.1, “Protocol over UNIS-6 and UNIS-12.” The FDT shall comply to the schema defined in Annex O, “FDT Schema Extensions”.

14.1.2 Network Generated Notification

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.

14.1.2.1 Protocol over UNIT-19

The use of FLUTE SHALL comply with [RFC3926]. 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 [RFC3926] with the extension defined in section 8.3.1.1, “Protocol over UNIS-6 and UNIS-12.” The FDT shall comply to the schema defined in Annex O, “FDT Schema Extensions”.

15 Diameter

This section defines the protocol for the use of Diameter protocol [RFC3588] over following referenced points.

- NPI-11

15.1 Protocols for IPTV Service Functions

15.1.1 Purchase of Digital Media

15.1.1.1 Protocol over NPI-11

The use of Diameter protocol SHALL comply with [DIAMCHG] and [RFC3588]. Two messages will be used, Accounting-Request (ACR) message and Accounting-Answer (ACA) message. The ACR message format is defined according to [RFC3588] (see section 6.2.2 of [DIAMCHG] to get more information about ACR). The ACA message format is defined according to [RFC3588] (see section 6.2.3 of [DIAMCHG] 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 section 6.2.2 of [DIAMCHG].

Step 2: The Charging FE returns ACA message to IPTV Control FE with purchase result. The ACA message is defined in section 6.2.3 of [DIAMCHG].

Annex A Change History (informative)

This Annex is intentionally left blank.

Annex B Example Messages (informative)

B.1 IPTV Service Functions Message Examples

B.1.1 Example Messages for unicast content streaming session setup with SIP session management

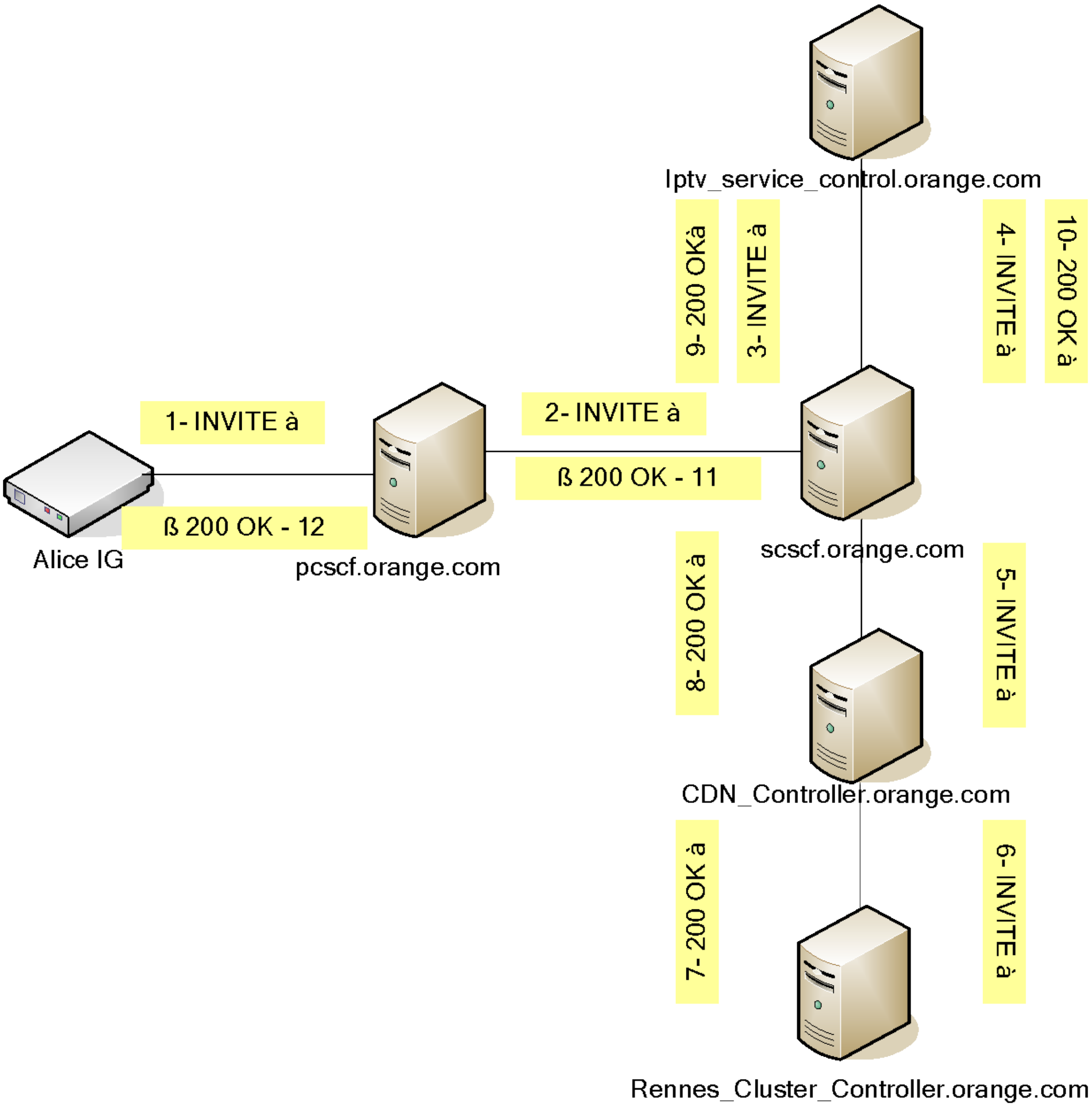


Figure 11: COD Session Set Up Sequence

Note: The IG had received in the response to the REGISTER the service route e.g.
<sip:pcscf.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

B.2 Communication Services Message Examples

B.2.1 Examples of HNI-IGI Message mapping to SIP

The sample mappings provided in this section are focused on Chat and Presence. They are not exhaustive.

The main use cases described in the Open IPTV Functional Architecture document 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 at the end of the section.

B.2.1.1 Presence

B.2.1.1.1 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:	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

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>	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

Notes:

Specifying both a ‘From’ and a ‘To’ allows an identity to publish on behalf of another identity

Here the server has requested a shorter expiration time that will be handled internally by the IG

B.2.1.1.2 Updated publication

HNI-IGI Interface	**SIP equivalent**
POST *IG_URI*/SIP X-OITF-Request-Line: PUBLISH sip:david@oiptv.org	PUBLISH sip:david@oiptv.org SIP/2.0 Via: SIP/2.0/UDP

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>	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

Notes:

SIP-IF-Match As retrieved from the previous publication acknowledgment

B.2.1.1.3 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/pidf+xml X-OITF- Content-length: 0 Content-Type: application/pidf+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/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: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

Notes:

Sip-if-match as retrieved from the previous publication acknowledgment

B.2.1.1.4 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/pidf+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/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: 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

Notes:

Here the server has requested a shorter expiration time that will be handled internally by the IG

B.2.1.1.5 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 that an acknowledgment from the IG to the network is REQUIRED but not described here.

B.2.1.1.6 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	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:

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	<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 that from the OITF’s point-of-view, both initial and refresh subscription requests are identical.

Here the server has requested a shorter expiration time, which will be handled internally by the IG

B.2.1.1.7 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	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:

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	<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

B.2.1.2 Chat

B.2.1.2.1 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	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

X-OITF-Content-Type: application/sdp	m=message 7654 TCP/MSRP *
X-OITF-Content-length: (...)	a=accept-types:message/cpim
	a=path:msrp://10.194.52.13:7654/jshA7weztas;tcp
Content-length: 0	

HTTP/1.1 200 OK	SIP/2.0 200 OK
X-OITF-Response-Line: SIP/2.0 200 OK	To: sip:sports.room@chat.oiptv.org;tag=087js
X-OITF-From: sip:david@oiptv.org	From: sip:david@oiptv.org;tag=786
X-OITF-To: sip:sports.room@chat.oiptv.org	Call-ID: 3413an89KU
X-OITF-Call-ID: 3413an89KU	Content-Type: application/sdp
X-OITF-Content-Type: application/sdp	Content-length: (...)
X-OITF-Accept: message/cpim	c=IN IP4 chat.oiptv.org
X-OITF-Content-length: (...)	m=message 12763 TCP/MSRP *
	a=accept-types:message/cpim
Content-Length: 0	a=path:msrp://chat.oiptv.org:12763/kjhd37s2s20w2a;tcp

Note that a final acknowledgment from the IG to the network is REQUIRED, but not described here.

B.2.1.2.2 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@oiptv.org X-HNI-IGI-To: sip:sports.room@chat.oiptv.org Who else thinks this late penalty was a disgrace?	MSRP a786hjs2 SEND To-Path: msrp://chat.oiptv.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.oiptv.org From: David <sip:david@oiptv.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@oiptv.org X-HNI-IGI-To: sip:sports.room@chat.oiptv.org Content-Length: 0	MSRP a786hjs2 200 OK To-Path: msrp://chat.oiptv.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

B.2.1.2.3 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@oiptv.org X-HNI-IGI-To: sip:sports.room@chat.oiptv.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.oiptv.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.oiptv.org From: David <sip:david@oiptv.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@oiptv.org X-HNI-IGI-To: sip:sports.room@chat.oiptv.org Content-Length: 0	MSRP a786hjs2 200 OK To-Path: msrp://chat.oiptv.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

B.2.1.2.4 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 that 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

B.2.1.2.5 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

B.2.1.3 Presence Document

B.2.1.3.1 Presence Schema

See Annex H for the Presence XML Schema.

B.2.1.3.2 Presence Schema examples

Examples of how the Presence information semantics are described in a typical Presence Information XML schema are shown below.

B.2.1.3.2.1 Example of Open IPTV 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>BCC</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 C User Profile Description (informative)

C.1 IPTV Subscription Profile

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 Schedueld Content service packages.

Broadcast service refers to Scheduled Content services. Accordingly, Broadcast settings refer to the Scheduled Content settings.

A Schedueld Content service package is a set of elementary Schedueld Content TV services, along with a description. These Schedueld Content services have the same authorization and charging policy.

A Schedueld Content IPTV service is for instance a multicast IPTV channel, interactive channel, mosaic that a user MAY subscribe to.

NOTE: The Broadcast settings only provide a reference to service package and/or associated services that a given IPTV user has subscribed to, 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 that detailed information about the OITF MAY be located elsewhere and can be referenced by the IPTV Subscription Profile using the tUEID element.

C.1.1 XML Schema for the IPTV Subscription Profile

XML Schema for the IPTV profile, based on [TS183063] 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>

  <xs:simpleType name="tBCServicePackageID" final="list restriction">
    <xs:restriction base="xs:string">
      <xs:minLength value="0" />
      <xs:maxLength value="16" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tBCServicePackageDescription" final="list restriction">
    <xs:restriction base="xs:string">
      <xs:minLength value="0" />
      <xs:maxLength value="64" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tBCService">
    <xs:sequence>
      <xs:element name="ParentalControl" type="tns:tParentalControlLevel"
        minOccurs="0" />
      <xs:element name="BCServiceId" type="tns:tBCServiceID" minOccurs="1" />
      <xs:element name="QualityDefinition" type="tns:tQualityDefinition"
        minOccurs="0" />
      <xs:element name="Extension" type="tns:tExtension" minOccurs="0" />
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="tBCServiceID" final="list restriction">
    <xs:restriction base="xs:string">
      <xs:minLength value="0" />
      <xs:maxLength value="16" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tQualityDefinition" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="1" />
      <xs:enumeration value="0">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">SD</xs:label>
            <xs:definition xml:lang="en">Standard Definition</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="1">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">HD</xs:label>
            <xs:definition xml:lang="en">High Definition</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tCoDProfile">
    <xs:sequence>
      <xs:element name="ParentalControl" type="tns:tParentalControlLevel"
        minOccurs="0"/>
      <xs:element name="Extension" type="tns:tExtension" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="tParentalControlLevel" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="5"/>
      <xs:enumeration value="0">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">ALL</xs:label>
            <xs:definition xml:lang="en">All contents</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="1">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">Level 1</xs:label>
            <xs:definition xml:lang="en">Level 1 contents</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="2">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">Level 2</xs:label>
            <xs:definition xml:lang="en">Up to level 2</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="3">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">Level 3</xs:label>
            <xs:definition xml:lang="en">Up to level 3</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="4">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">Level 4</xs:label>
            <xs:definition xml:lang="en">Up to level 4</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="5">
        <xs:annotation>
          <xs:documentation>
            <xs:label xml:lang="en">Level 5</xs:label>
            <xs:definition xml:lang="en">Up to level 5</xs:definition>
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tPVRProfile">
    <xs:sequence>
      <xs:annotation>
        <xs:documentation>
          Unit of the StorageLimitInVolume element is the GigaOctet
        </xs:documentation>
      </xs:annotation>
      <xs:element name="PVRPreference" type="tns:tPVRPreference"/>
      <xs:element name="StorageLimitInTime" type="tns:tStorageLimitInTime"
        minOccurs="0"/>
      <xs:element name="StorageLimitInVolume" type="tns:tStorageLimitInVolume"
        minOccurs="0"/>
      <xs:element name="Extension" type="tns:tExtension" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="tPVRPreference" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
      <xs:enumeration value="0">
```

```
      <xs:annotation>
        <xs:documentation>
          <xs:label xml:lang="en">Network</xs:label>
          <xs:definition xml:lang="en">
            Recording is done in the network
          </xs:definition>
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="1">
      <xs:annotation>
        <xs:documentation>
          <xs:label xml:lang="en">User_Equipment</xs:label>
          <xs:definition xml:lang="en">
            Recording is done on the user equipment
          </xs:definition>
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tNPVRStorageLimitInTime">
  <xs:restriction base="xs:duration">
    <xs:minInclusive value="PT0H"/>
    <xs:maxInclusive value="PT1000000000H"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tNPVRStorageLimitInVolume">
  <xs:restriction base="xs:nonNegativeInteger"/>
</xs:simpleType>

<xs:complexType name="tGlobalSettings">
  <xs:sequence>
    <xs:element name="LanguagePreference" type="tns:tLanguage" minOccurs="0"/>
    <xs:element name="UsersActionRecodable" type="tns:tUserActionRecordable"/>
    <xs:element name="Extension" type="tns:tExtension" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="tLanguage">
  <xs:restriction base="xs:string">
    <xs:annotation>
      <xs:documentation>
        <xs:definition xml:lang="en">ISO 639-2 Language code</xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:minLength value="3"/>
    <xs:maxLength value="3"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="tExtension">
  <xs:sequence>
    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="tUserActionRecordable">
```

```
    <xs:restriction base="xs:boolean"/>
  </xs:simpleType>

</xs:schema>
```

C.2 XML Schema for the OIPF Profile

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="org:oipf:iptv:UEProfile:2010"
  xmlns:tns="org:oipf:itpv:UEProfile: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:annotation>
    <xs:documentation xml:lang="en">
      Defines the capabilities of the OIPF that is currently
      associated with the user
    </xs:documentation>
  </xs:annotation>

  <xs:element name="UEInformation" type="tns:tUEProfile" />
  <xs:complexType name="tUEProfile">
    <xs:sequence>
      <xs:element name="UserEquipmentID" type="tns:tUEID" />
      <xs:element name="UserEquipmentClass" type="tns:tUserEquipmentClass" />
      <xs:element name="Resolution" type="tns:tResolution"
        minOccurs="0" />
      <xs:element name="SupportedEncodings" type="tns:tSupportedEncodings"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="SupportedProtocols" type="tns:tSupportedProtocols"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="SupportedContentProtection"
        type="tns:tSupportedContentProtection"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="SupportedCSPG" type="tns:tCSPG"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="IPEncapsulations" type="tns:tIPEncapsulations"
        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>

  <xs:simpleType name="tUEID">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">User Equipment ID</xs:label>
        <xs:definition xml:lang="en">
          Unique Identifier for the OIPF(to be specified)
        </xs:definition>
      </xs:documentation>
    </xs:annotation>

    <xs:restriction base="xs:string">
      <xs:minLength value="0" />
      <xs:maxLength value="16" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tUserEquipmentClass"
    final="list restriction">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">User Equipment class</xs:label>
        <xs:definition xml:lang="en">
          Specifies the type of OIPF
        </xs:definition>
      </xs:documentation>
    </xs:annotation>

    <xs:restriction base="xs:string">
      <xs:enumeration value="OITF-TV" />
      <xs:enumeration value="OITF-STB" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tResolution">
    <xs:attribute name="HorizontalSize" type="xs:integer">
      <xs:annotation>
        <xs:documentation>
          horizontal size in pixels of the screen
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="VerticalSize" type="xs:integer">
      <xs:annotation>
        <xs:documentation>
          vertical size in pixels of the screen
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Rotate" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          set to TRUE if the screen can be rotated (horizontal
          becomes vertical)
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="tSupportedEncodings">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">encodings</xs:label>
        <xs:definition xml:lang="en">
          Specifies the supported audio and video encodings
          (eg. MPEG2,H264 AC3, AAC etc)
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="AudioEncoding" type="tns:tAudioEncoding"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="VideoEncoding" type="tns:tVideoEncoding"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tAudioEncoding">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">Audio Encoding</xs:label>
        <xs:definition xml:lang="en">
          Specifies supported audio encoding Properties
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Encoding" type="tva:ControlledTermType" />
      <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="tVideoEncoding">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">Video Encoding</xs:label>
        <xs:definition xml:lang="en">
          Specifies supported video encoding properties
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Encoding" type="tva:ControlledTermType" />
      <xs:element name="SupportedFrameRate" type="tva:FrameRateType"
        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>

  <xs:simpleType name="tSupportedProtocols">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">Protocols</xs:label>
        <xs:definition xml:lang="en">
          Specifies a list of supported protocols ('OIPF-HTTP-AS' for HTTP
          Adaptive Streaming as defined in OIPF, 'HTTP-PDL' for Progressive
          Download, 'HTTP-DL' for HTTP Download)
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="OIPF-HTTP-AS" />
      <xs:enumeration value="HTTP-PDL" />
      <xs:enumeration value="HTTP-DL" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tSupportedContentProtection">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">Content Protection</xs:label>
        <xs:definition xml:lang="en">
          Specifies the supported content protection system (e.g.,
          "urn:dvb:casystemid:19188") with optionally the gateway (e.g., "CI+" or
          "DTCP-IP") and supported protected formats
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="ProtectedFormat" type="tProtectedFormat"
        minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##any" processContents="lax"
          minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="ContentProtectionSystemID"
      type="xs:string" use="required" />
    <xs:attribute name="CSPG" type="tns:tCSPG" use="optional" />
    <xs:anyAttribute namespace="##any" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:complexType>

  <xs:simpleType name="tCSPG">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">CSPG type</xs:label>
        <xs:definition xml:lang="en">
          Specifies the type of CSPG
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="OIPF-CI+" />
      <xs:enumeration value="OIPF-DTCP-IP" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tProtectedFormat">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">Protected Format</xs:label>
        <xs:definition xml:lang="en">
          Specifies the supported Protected Format
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="BBTS" />
      <xs:enumeration value="PF" />
      <xs:enumeration value="PDCF" />
      <xs:enumeration value="MPIMP" />
      <xs:enumeration value="DCF" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tIPEncapsulations" final="list restriction">
    <xs:annotation>
      <xs:documentation>
        <xs:label xml:lang="en">encapsulation</xs:label>
        <xs:definition xml:lang="en">
          Specifies the IP encapsulation that is supported on
          the device (UDP/RTP, UDP/M2TS, UDP/RTP/M2TS)
        </xs:definition>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="UDP/RTP" />
      <xs:enumeration value="UDP/M2TS" />
      <xs:enumeration value="UDP/RTP/M2TS" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tExtension">
    <xs:sequence>
      <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

C.3 IPTV Subscription Profile Elements classification

In this section, IPTV Subscription Profile elements are classified according to the desired visibility to the user:

- User visible and manageable data
- User visible, but not user-manageable data
- Data neither visible nor manageable by the user (service parameters that remain in the network, etc.)

Elements MAY remain without classification (that would mean that it is not determined if the data SHOULD remain in the network or visible to the user.)

element="IPTVProfile"

attribute ="ProfileId"

C.3.1 User visible and manageable data

CoD Profile:

In complexType ="tCoDProfile"

element ="ParentalControl" type="tParentalControlLevel

Global Settings:

In complexType ="tGlobalSettings"

element="LanguagePreference" type=tLanguage

C.3.2 User visible, but not manageable data

The term UE (User Equipment) is equivalent to the Open IPTV Forum term OITF.

In complexType="tUEProfile"

element ="UserEquipmentID" type="tUEID"

element ="UECapabilities" type="tUECapabilities"

In complexType="tUECapabilities">

element ="UserEquipmentClass" type="UserEquipmentClass"

N-PVR:

In complexType="tPVRProfile"

element="PVRPreference" type ="tPVRPreference"

element ="StorageLimitInTime" type="tNPVRStorageLimitInTime"

element ="StorageLimitInVolume" type="tNPVRStorageLimitInVolume"

C.3.3 Data neither visible nor manageable by the user

BC Profile:

In complexType ="tBCProfile"

element ="BCServicePackage" type="tBCServicePackage"

In complexType ="tBCServicePackage"

element name="BCPackageId" type="tBCServicePackageID"

element name="Description" type="tBCServicePackageDescription"

element name="BCService" type="tBCService"

In complexType ="tBCService"

element name="BCServiceId" type="tBCServiceID"

element name="QualityDefinition" type="tQualityDefinition"

Annex D Mapping attributes for Scheduled Content

D.1 Mapping SDP attributes from DVB SD&S information

IP SDP parameters for each media stream	**Corresponding DVB SD&S element in [TS102034] section 5.2.6.2 tables 4, 5 and 8.**
Scheduled Content stream	
Connection Data c=<network type> <address type> <connection address>	
<network type>	Not retrieved from SD&S
<address type>	Not retrieved from SD&S
<connection address>	IPMulticastAddress@Address
Media Announcements for content delivery m=<media> <port> <proto> <fmt >	
<media>	"video" (also present in SD&S)
<port>	IPMulticastAddress@Port
<proto>	“RTP/AVP” if IPMulticastAddress@Streaming=“rtp” or if IPMulticastAddress@Streaming is not present
	“RTP/AVPF” if (IPMulticastAddress@Streaming=”rtp” or if IPMulticastAddress@Streaming is not present) and OITF wants to make use of FCC and/or RET service offer. (Note 1)
	“MP2T/H2221/UDP” or “RAW/RAW/UDP” if IPMulticastAddress@Streaming=“udp”
<fmt>	When MPEG2-Transport Stream [MPEG2TS] is used, <fmt> SHALL be “33” as specified in RFC 3551 [RFC3551]. When OPTIONAL Timestamped-TS defined by [DLNA] 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 [DLNA]. Example m=video 49232 RTP/AVP 98 a=rtpmap:98 vnd.dlna.mpeg-tts/27000000
Bandwidth b=AS:<bandwidth>	MaxBitrate Note 1: the “MaxBitrate” sttribute in SD&S is calculated according to the TIAS bandwidth modifier defined in RFC 3890 [RFC3890], but expressed in kbps. The OITF SHOULD do the necessary conversion to express the bandwidth in the SDP as “b=AS:<bandwidth>”. Note 2: If the OITF wants to make use of FCC/RET service, the <bandwidth> for the FCC/RET enabled multicast content service is to be retrieved from the SIP OPTIONS response message.
BCServiceId	TextualIdentifier@ServiceName“:”TextualIdentifier@DomainName Note that the TextualIdentifier@DomainName is an OPTIONAL attribute; therefore when not present, it is copied from the OfferingBase@DomainName
BCPackageId	Package@Id

FEC stream Note that the multicast address and source address of the FEC stream can be the same as the multicact content stream.	
Media Announcements for FEC delivery m=<media> <port> <proto> <fmt>	Note: the FEC delivery can only be associated to a RTP delivered content.
<media>	“application”, not retrieved from SD&S
<port>	IPMulticastAddress.FECBaseLayer@Port
<proto>	RTP/AVP
<fmt>	Dynamic payload type
a=rtpmap:<fmt> <encoding_name/clock_rate>	<encoding_name/clock_rate> referring to the DVB-IP AL-FEC Base layer and is equal to: “vnd.dvb.iptv.alfec-base/90000”
Connection Data at media level c=<network type> <address type> <connection address>	
<network type>	Not retrieved from SD&S
<address type>	Not retrieved from SD&S
<connection address>	IPMulticastAddress.FECBaseLayer@Address
Network Generated Notification stream and Emergency Notification stream Note that the multicast address and source address of the notification stream can be the same as the multicact content stream.	
Connection Data at media level c=<network type> <address type> <connection address>	
<network type>	Not retrieved from SD&S
<address type>	Not retrieved from SD&S
<connection address>	IPMulticastAddress@Address
Media Announcements for content delivery m=<media> <port> <proto> <fmt >	
<media>	"application" (also present in SD&S)
<port>	IPMulticastAddress@Port
<proto>	“FLUTE/UDP”
<fmt>	0
BCServiceId	TextualIdentifier@ServiceName“:”TextualIdentifier@DomainName Note that the TextualIdentifier@DomainName is an OPTIONAL attribute; therefore when not present, it is copied from the OfferingBase@DomainName
BCPackageId	Package@Id
IP multicast RET stream Note that the multicast address and source address of the multicast RET stream can be the same as the multicast content stream.	
Media Announcements for multicast RET delivery m=<media> <port> <proto> <fmt >	Note: The multicast RET delivery can only be associated to RTP delivered content
<media>	"application", not retrieved from SD&S
<port>	MulticastRET@DestinationPort
<proto>	“RTP/AVP”
<fmt>	MulticastRET@RTPPayloadTypeNumber
a=rtpmap:<fmt> <encoding_name/clock_rate>	<encoding_name/clock_rate> is equal to:

	“rtx/90000” [RFC4588] (not signalled in SD&S)
Connection Data at media level c=<network type> <address type> <connection address>	
<network type>	Not retrieved from SD&S
<address type>	Not retrieved from SD&S
<connection address>	MulticastRET@GroupAddress

Note: RTP/AVPF [RFC4588] is the protocol identifier by which an RTP agent can indicate it supports the extended RTP profile for RTCP-based feedback. It is REQUIRED for FCC/RET services.

D.2 Service Package SDP attributes

The format of the a=bc_service_package attribute is the following:

```
a= bc_service_package : <BCPackageId> [mult_list] [bc_tv_service_id_list]

where

        <mult_list> ::= mult_list:<source_unit>{“|”<source_unit>}

        <source_unit> ::= [src_list:”(“<src-list>"),"]<multicast_address>{(“,”|”-“)<multicast_address>}

        <src-list> ::= <source_address>{(“,”|”-“)<source_address>}

        <source_address> ::= <IP_address>

        <multicast_address> ::= <IP_address>

        <bc_tv_service_id_list>::=<BCServiceId> {“,”<BCServiceId>}

        <BCServiceId> is the string defined above.

        <BCPackageId> is the service package ID string defined above.
```

(BNF notation). As seen in this notation the multi_list parameter can contain one or more source_unit parameters with multicast addresses that can be separated with either ‘,’ or ‘-’.

When they are separated with ‘-’ it means that it is a range of addresses. In addition there can OPTIONALly be a list of source addresses within the source unit parameter which is applicable for all the multicast addresses within the source unit parameter.

Annex E <protocol> names

E.1 Definition of <protocol> names

Following table shows the names (labels) of <protocol> which SHALL be a combination of signalling protocols and media transport protocols on UNI.

Table 142: Definition of <protocol> names

Service	**Signalling protocol**	**Media Transport protocol**	**Name of <protocol>**
Multuicast content streaming	SIP + IGMP	RTP	“sip-igmp-rtp-udp”
		direct-UDP	“sip-igmp-udp”
	IGMP	RTP	“igmp-rtp-udp”
	N/A	HTTP Adaptive Streaming	“dash”
Unicast content streaming	SIP + RTSP	RTP	“sip-rtsp-rtp-udp”
		direct-UDP	“sip-rtsp-udp”
	RTSP	RTP	“rtsp-rtp-udp”
	N/A	HTTP	“http-get”
	N/A	HTTP Adaptive Streaming	“has”
	N/A	HTTP Adaptive Streaming	“dash”
Content download	N/A	HTTP	“http-get”

Note 1: SIP based signalling protocols can only be used in managed network relying on IMS.

Annex F System Infrastructure

F.1 OITF Start up High-Level Procedures

F.1.1 OITF with Native HNI-IGI Support

Figure 12 shows the high-level procedural flow for OITF starts up i.e. up to the point where all OITF functions are available. The following is a description of the steps:

Step 1: The local device start up procedure (which is implementation dependent).

Step 2: The OITF SHALL discover the IG through a UPnP procedure (section 10.1.1.1, “Procedure for IG Discovery”).

Step 3: The OITF SHALL use the DHCP option 124/125 to query the DHCP server to obtain the SP Discovery entry point (see section 12.1.1.1.3, “Option 124/125”.) If the deployment includes an IG, the DHCP server SHOULD[1] be configured to return the IG address in the DHCP option 125, either as a FQDN or as an IP address. In other words, in such a deployment the IG acts as the SP Discovery entry point (see Annex F.4, “IG Startup and Shutdown procedures” for how an IG acts in this role).

Step 4: The OITF SHALL retrieve the list of subscription identities (IMPUs) (section 5.4.6.3, “HTTP Digest Authentication.”)

Step 5: The OITF SHALL registers the user identity with the IG (section 5.4.6.1, “Procedure for User Registration and Authentication in network relying on IMS on the HNI-IGI Interface.”)

Step 6: If the IG supports GBA, the OITF SHALL perform GBA authentication (section 5.4.6.2.1, “Initial GBA registration.”)

Step 7: The OITF SHALL perform service provider discovery (section 5.4.1, “Service Provider Discovery.”) The service provider information MAY be returned directly by the IG.

Step 8: The OITF (or the DAE application, whichever applies) SHALL prompt the user to choose an SP. For any device, the timing and method of presentation as well as the relative positioning of the different SPs to the user is out of scope of the IPTV Solution specifications.

Step 9: The OITF (or the retrieved DAE application, whichever applies) SHALL perform service discovery (see section 5.4.2, “Service Discovery”.)

[1] The DHCP server MAY, for example, return the FQDN or IP address of the SP Discovery FE in the network instead.

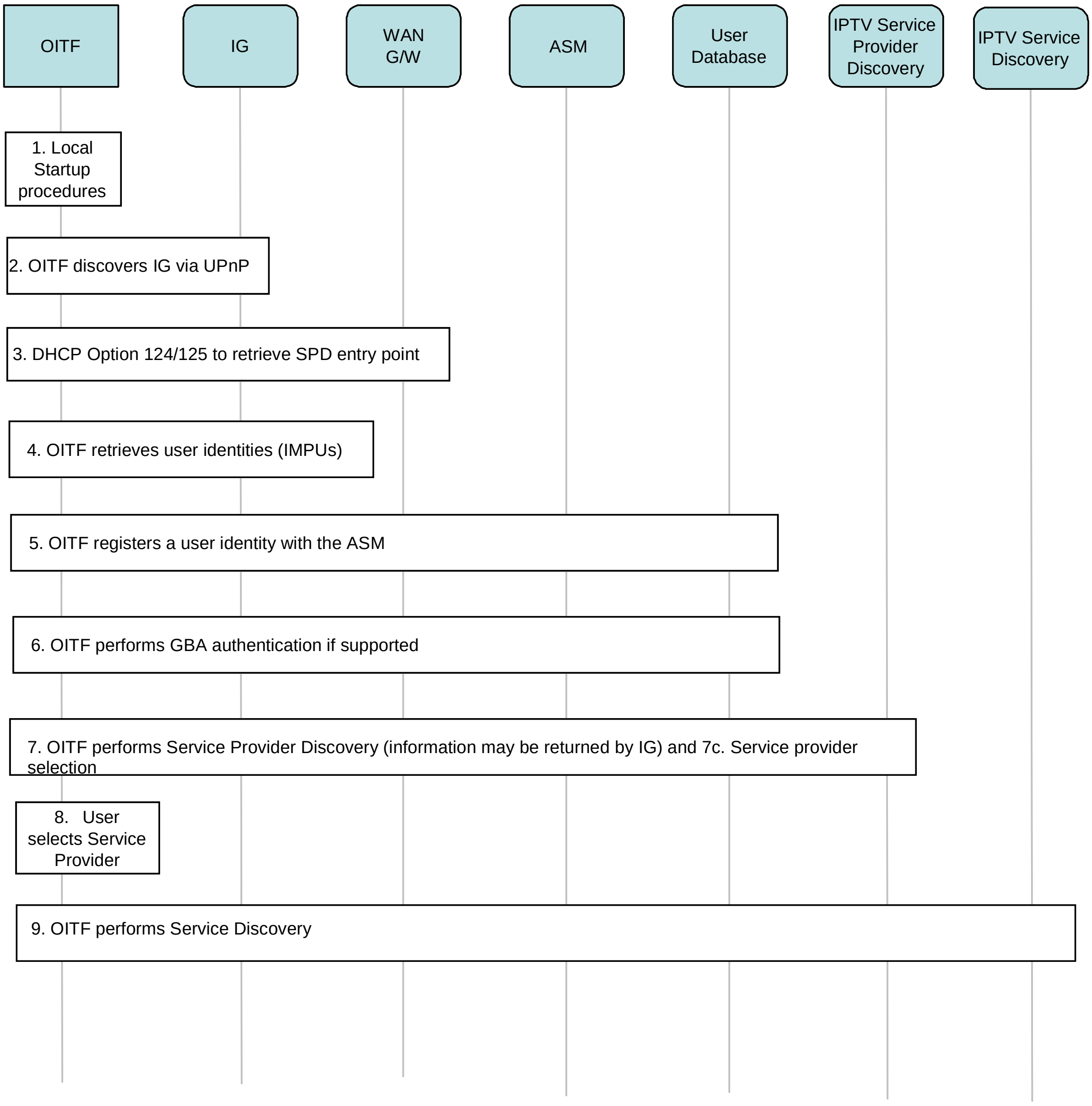

OITF
IG
WAN G/W
ASM
User Database
IPTV Service Provider Discovery
IPTV Service Discovery
1. Local Startup procedures
2. OITF discovers IG via UPnP
3. DHCP Option 124/125 to retrieve SPD entry point
4. OITF retrieves user identities (IMPUs)
5. OITF registers a user identity with the ASM
6. OITF performs GBA authentication if supported
7. OITF performs Service Provider Discovery (information may be returned by IG) and 7c. Service provider selection
8. User selects Service Provider
9. OITF performs Service Discovery

Figure 12: High level Start up procedural flow for an OITF with native HNI-IGI support

F.1.2 OITF with Non-Native HNI-IGI Support

Step 1: The local device start up procedure (which is implementation dependent).

Step 2: The OITF SHALL use the DHCP option 124/125 to query the DHCP server to obtain the SP Discovery entry point (see section 12.1.1.1.3, "Option 124/125".) If the deployment includes an IG, the DHCP server SHOULD[2] be configured to return the IG address in the DHCP option 125, either as a FQDN or as an IP address. In other words, in such a deployment the IG acts as the SP Discovery entry point (see Annex F.4, "IG Startup and Shutdown procedures" for how an IG acts in this role).

Step 3: The OITF that has received the SP Discovery entry point via DHCP option 125 SHALL retrieve the Service Provider information by querying this entry point (see section 5.4.1.2, "Protocol over UNIS-19 and Non-native HNI-IGI.") For any device, the time at which to trigger the query for Service Provider Discovery information is out of scope of the IPTV Solution specifications.

[2] The DHCP server MAY, e.g., return the FQDN or IP address of the SP Discovery FE in the network instead.

Step 4: The OITF (or the DAE application, whichever applies) SHALL prompt the user to choose an SP. For any device, the timing and method of presentation as well as the relative positioning of the different SPs to the user is out of scope of the IPTV Solution specifications.

Step 5: The OITF SHALL retrieve the list of user identities from the IG using the DAE application retrieved in step 4 (see section 5.4.6.3, “HTTP Digest Authentication”.)

Step 6: The OITF SHALL register a user identity with the service platform provider, using a DAE application retrieved in step 4 (see section 5.4.6.1, “Procedure for User Registration and Authentication in network relying on IMS on the HNI-IGI Interface”.)

Step 7: The OITF (or the retrieved DAE application, whichever applies) SHALL perform service discovery (see section 5.4.2, “Service Discovery”.)

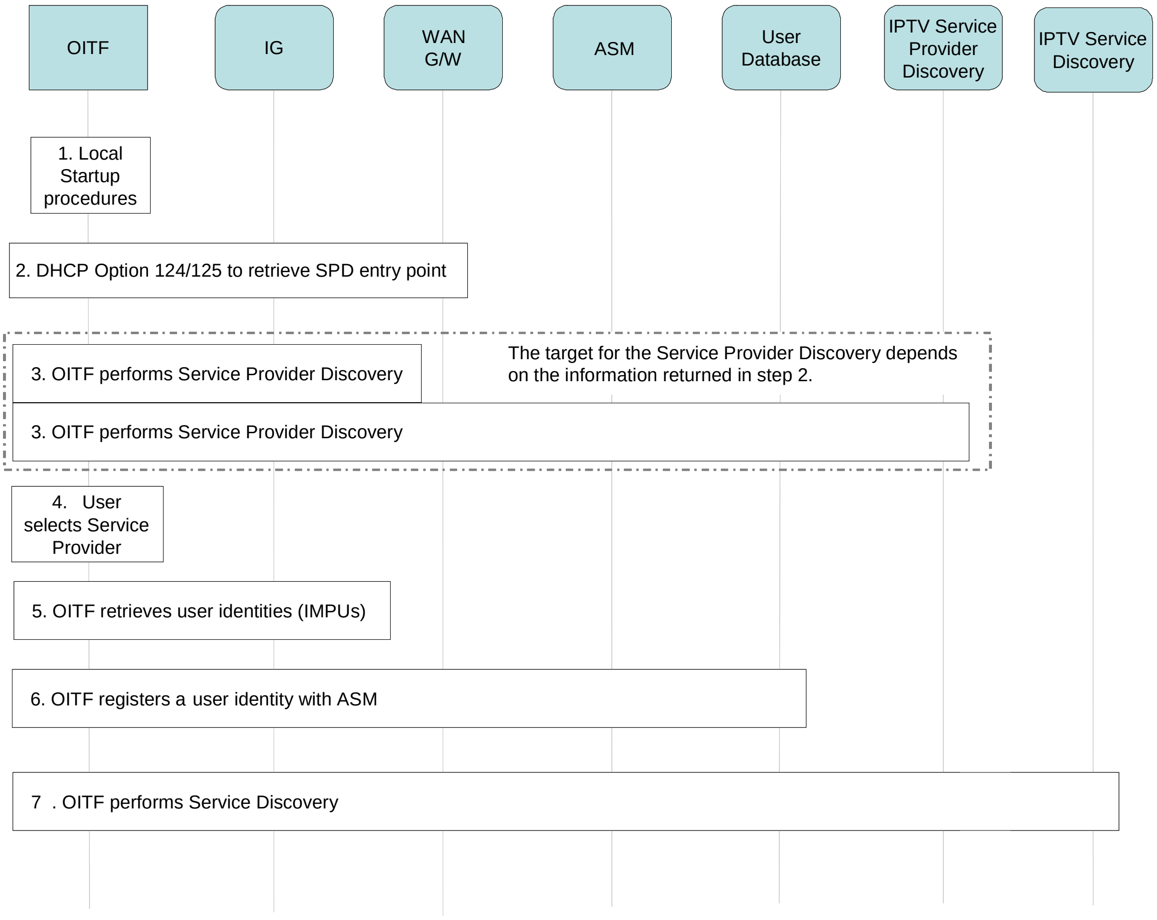

OITF
IG
WAN G/W
ASM
User Database
IPTV Service Provider Discovery
IPTV Service Discovery
1. Local Startup procedures
2. DHCP Option 124/125 to retrieve SPD entry point
3. OITF performs Service Provider Discovery
The target for the Service Provider Discovery depends on the information returned in step 2.
3. OITF performs Service Provider Discovery
4. User selects Service Provider
5. OITF retrieves user identities (IMPUs)
6. OITF registers a user identity with ASM
7 . OITF performs Service Discovery

Figure 13: High-level start-up procedural flow for an OITF without native HNI-IGI support

F.1.3 Integrated OITF/IG with no HNI-IGI Support

Figure 14 shows the high-level procedural flow for an integrated OITF/IG device with no HNI-IGI support . The following is a description of the steps:

Step 1: The local device start up procedure (which is implementation dependent).

Step 2: The IG SHALL register the default user identity (section 5.4.6.1, “Procedure for User Registration and Authentication in network relying on IMS on the HNI-IGI Interface.”).

Step 3: The IG SHALL perform GBA authentication if it supports the procedure (section 5.4.6.2.1, “Initial GBA registration.”)

Step 4a: The IG SHALL perform service provider discovery (section 5.4.1, “Service Provider Discovery.”).

Step 4b: The OITF (or the DAE application, whichever applies) SHALL prompt the user to choose an SP. For any device, the timing and method of presentation as well as the relative positioning of the different SPs to the user is out of scope of the IPTV Solution specifications.

Step 5: The OITF (or the retrieved DAE application, whichever applies) SHALL perform service discovery (see section 5.4.2, "Service Discovery".)

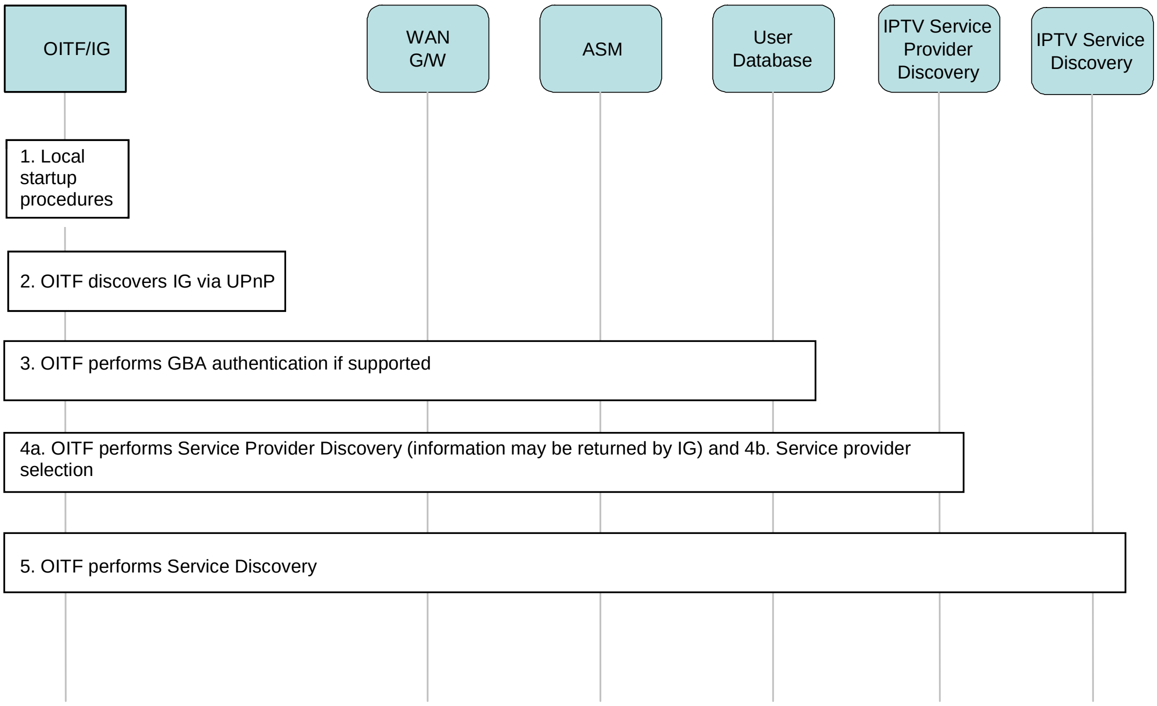

OITF/IG
WAN G/W
ASM
User Database
IPTV Service Provider Discovery
IPTV Service Discovery
1. Local startup procedures
2. OITF discovers IG via UPnP
3. OITF performs GBA authentication if supported
4a. OITF performs Service Provider Discovery (information may be returned by IG) and 4b. Service provider selection
5. OITF performs Service Discovery

Figure 14: High-level start-up procedural flow for an integrated OITF/IG

F.2 High-Level Procedure for an OITF Graceful Shut Down in network relying on IMS

Figure 15 shows a high-level procedural flow when an OITF is shut down, i.e. the OITF functionality is made completely inactive. The following steps are performed:

Step 1: The OITF gracefully terminates any ongoing IPTV session. (See appropriate IPTV Service Termination sections)

Step 2: The OITF de-registers the logged-in users (See section 5.4.6.1.2, "User De-registration.")

Step 3: The IG terminates all activities for communication services for the de-registered identity that are associated with the OITF contact point (IP address) (See section 6.1.4, "Protocols for Communication Services.")

Step 4: The IG de-registers the logged in user from the network. (See section 6.1.3.2.2, "Procedure for User Registration and Authentication in a network relying on IMS on UNIS-8.") If this is the last OITF to shut down and if this is a default identity, then the IG MUST deregister the old contact point with the network and re-register the IG as the new contact point. If this is a user identity being deregistered, then the IG MUST deregister this identity and register a default identity with the IG as the contact point.

Step 5: The OITF performs local shut down procedure.

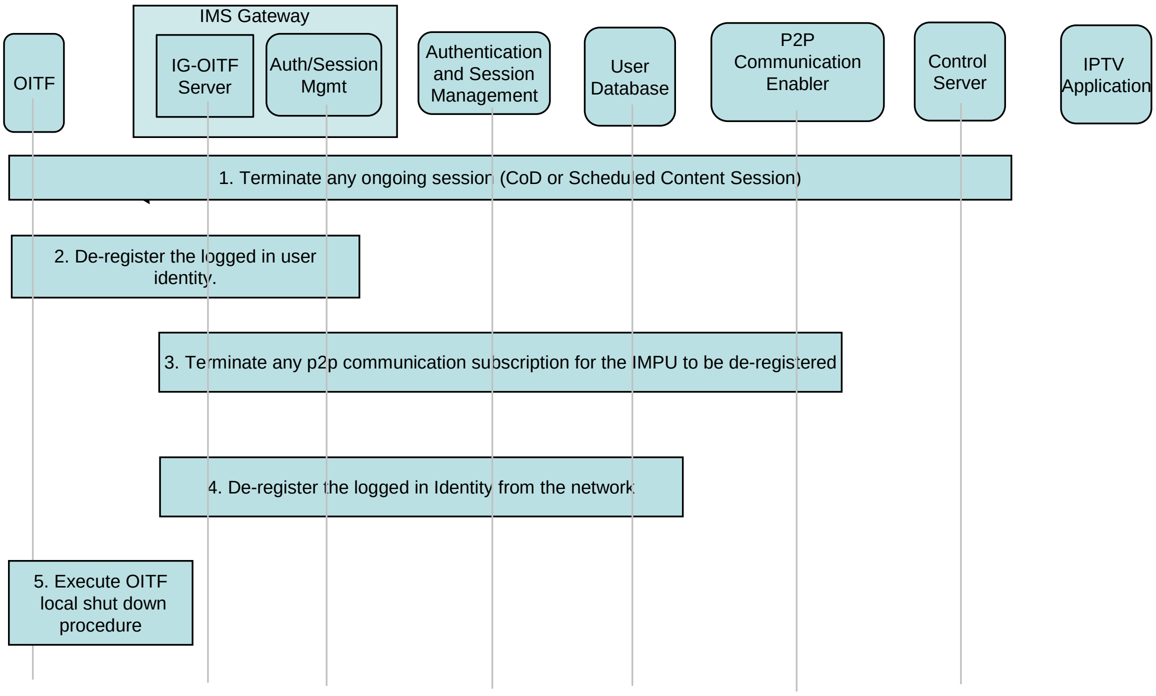

IMS Gateway
OITF
IG-OITF Server
Auth/Session Mgmt
Authentication and Session Management
User Database
P2P Communication Enabler
Control Server
IPTV Application
1. Terminate any ongoing session (CoD or Scheduled Content Session)
2. De-register the logged in user identity.
3. Terminate any p2p communication subscription for the IMPU to be de-registered
4. De-register the logged in Identity from the network
5. Execute OITF local shut down procedure

Figure 15: High level Shut-down procedural flow for an OITF

F.3 OITF Restart high level procedures for an IG integrating WAN GW

This section details how stale SIP state can be detected in an IG integrating the GW, i.e. IG-GW, when an OITF restarts. This procedure is valid for both native and non-native HNI-IGI interfaces.

Figure 16 depicts how the IG-GW is able to establish a mapping between the SIP state (SIP dialog, IMPU and IP address) and the network state (IP address and deviceID).

The ability of the IG-GW to detect stale SIP state upon restart is based on the fact that when an OITF restarts, it re-initiates the DHCP server discovery (sends a DHCPDISCOVER message) and IP address request (DHCPREQUEST) procedures. This is an indication that the OITF has re-started. This is depicted in Figure 17.

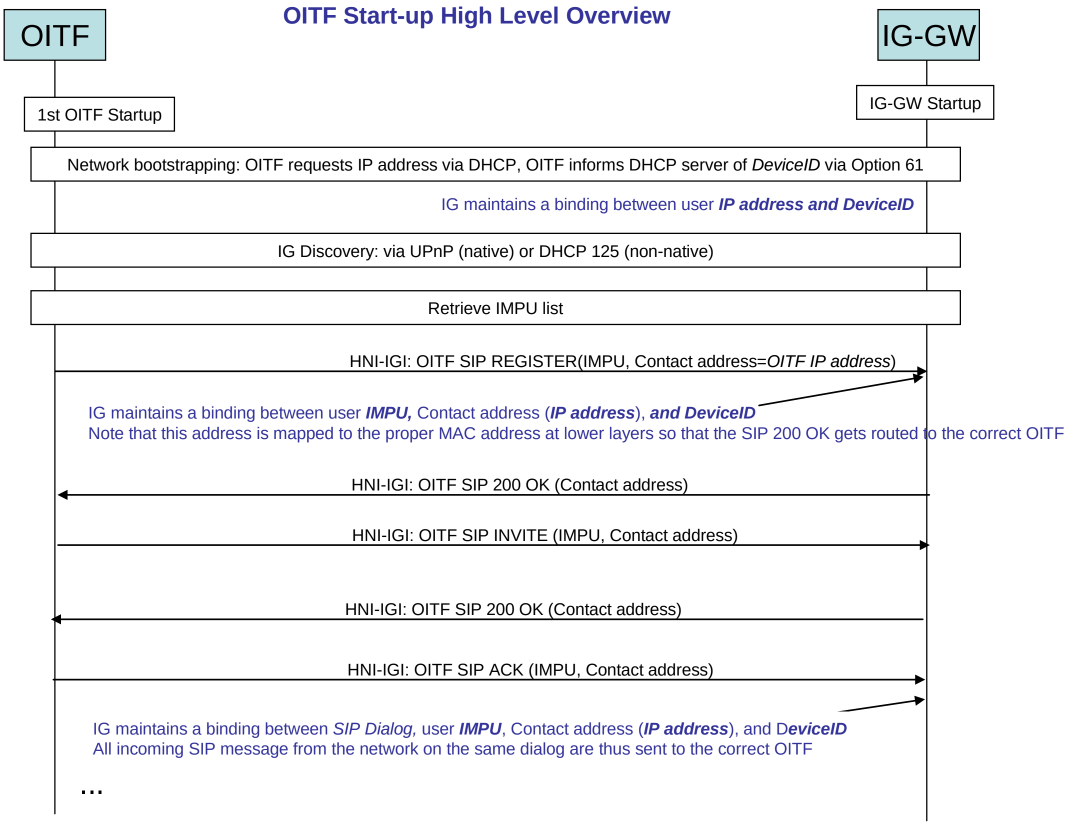

OITF Start-up High Level Overview
OITF
IG-GW
1st OITF Startup
IG-GW Startup
Network bootstrapping: OITF requests IP address via DHCP, OITF informs DHCP server of *DeviceID* via Option 61
IG maintains a binding between user ***IP address and DeviceID***
IG Discovery: via UPnP (native) or DHCP 125 (non-native)
Retrieve IMPU list
HNI-IGI: OITF SIP REGISTER(IMPU, Contact address=*OITF IP address*)
IG maintains a binding between user ***IMPU,*** Contact address (***IP address***), ***and DeviceID***
Note that this address is mapped to the proper MAC address at lower layers so that the SIP 200 OK gets routed to the correct OITF
HNI-IGI: OITF SIP 200 OK (Contact address)
HNI-IGI: OITF SIP INVITE (IMPU, Contact address)
HNI-IGI: OITF SIP 200 OK (Contact address)
HNI-IGI: OITF SIP ACK (IMPU, Contact address)
IG maintains a binding between *SIP Dialog,* user ***IMPU***, Contact address (***IP address***), and D***eviceID***
All incoming SIP message from the network on the same dialog are thus sent to the correct OITF
…

Figure 16: Overview of OITF Startup

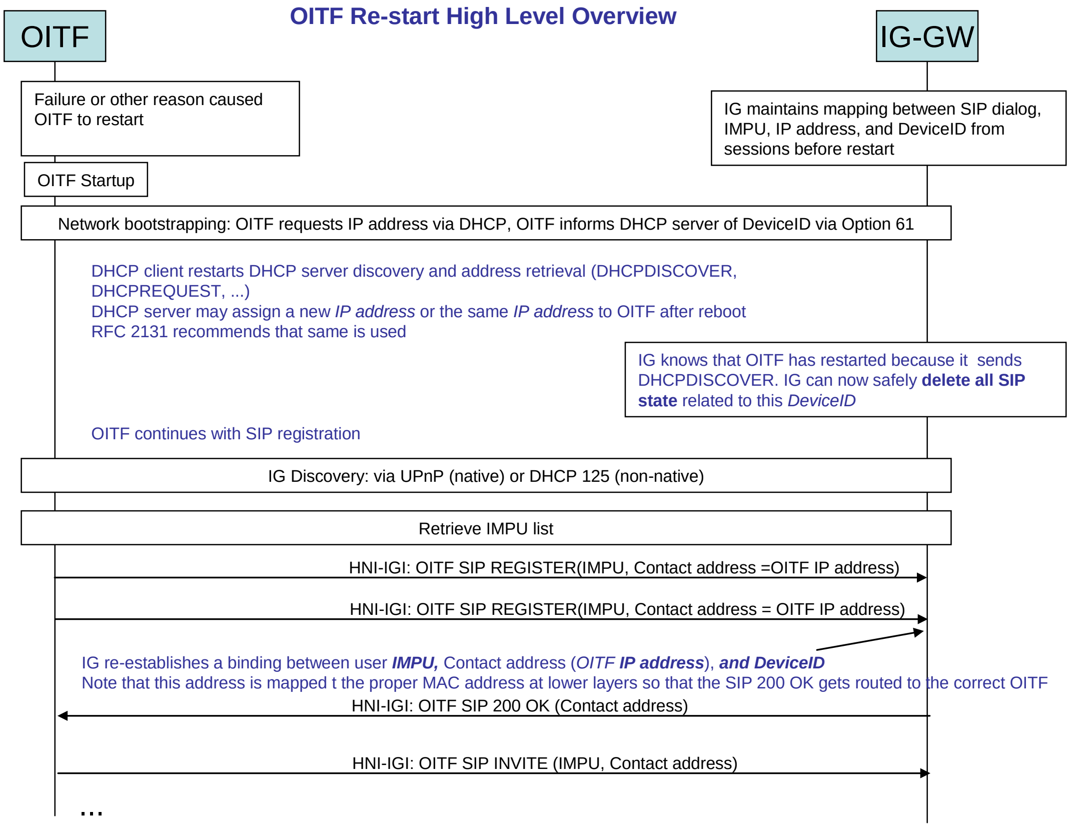

OITF Re-start High Level Overview
OITF
IG-GW
Failure or other reason caused OITF to restart
IG maintains mapping between SIP dialog, IMPU, IP address, and DeviceID from sessions before restart
OITF Startup
Network bootstrapping: OITF requests IP address via DHCP, OITF informs DHCP server of DeviceID via Option 61
DHCP client restarts DHCP server discovery and address retrieval (DHCPDISCOVER, DHCPREQUEST, ...)
DHCP server may assign a new *IP address* or the same *IP address* to OITF after reboot
RFC 2131 recommends that same is used
IG knows that OITF has restarted because it sends DHCPDISCOVER. IG can now safely **delete all SIP state** related to this *DeviceID*
OITF continues with SIP registration
IG Discovery: via UPnP (native) or DHCP 125 (non-native)
Retrieve IMPU list
HNI-IGI: OITF SIP REGISTER(IMPU, Contact address =OITF IP address)
HNI-IGI: OITF SIP REGISTER(IMPU, Contact address = OITF IP address)
IG re-establishes a binding between user ***IMPU,*** Contact address (*OITF* ***IP address***), ***and DeviceID***
Note that this address is mapped t the proper MAC address at lower layers so that the SIP 200 OK gets routed to the correct OITF
HNI-IGI: OITF SIP 200 OK (Contact address)
HNI-IGI: OITF SIP INVITE (IMPU, Contact address)
…

Figure 17: Overview of OITF Restart

F.4 IG Startup and Shutdown procedures

F.4.1 IG Startup procedures

Step 1: IG power up initialization procedures (this is implementation dependent).

Step 2: The IG SHALL retrieve or receive user identities associated with the IMS subscription and other configuration information from the network (see section 5.4.5.1.3, “Configuration of the IG via Configuration File”)

Step 3: The IG SHALL register the default identity associated with the IMS subscription with the service platform (IMS) provider (see user registration section 6.1.3.2)

Step 4: The IG SHALL perform the SP Discovery procedure (see section 6.1.3.1 for details). The IG SHALL store the SP Discovery information in the format (see [OIPF_META2]) it was received.

At this point, the IG has completed its startup procedures and is ready to accept requests from the OITF and/or network entities.

The IG SHALL drop any messages received from the network related to the default user until such time that it detects that a default user has registered at an OITF.

F.4.2 IG Shutdown procedures

Step 1: The IG terminates all activities for communication services.

Step 2: The IG SHALL terminate all other SIP sessions.

Step 3: IG SHALL de-register all users bound to OITFs.

Step 4: The IG SHALL de-register any IG-initiated registration (if applicable).

Step 5: The IG performs its local shutdown procedure (implementation dependent)

F.5 WAN Gateway Functions

The WAN Gateway SHALL support multiple in-home devices for the consumer network, which SHALL be able to join the same multicast streams.

It SHOULD support IGMP snooping on all LAN side interfaces and forward inbound multicast packets to those physical interfaces which are connected to devices that have joined the specific multicast group.

The WAN Gateway SHALL support full IGMP v3 ([RFC3376]).

It SHALL implement an IGMP proxy mechanism ([RFC4605]).

F.6 NAT Traversal

The reason why IPTV will not function by default behind a NAT is that many of the communication parameters in SIP and in RTSP are transported within the SDP message; these parameters include the IP and port numbers used for signalling and media. A device behind a NAT does not know how it will be seen from the Network domain; it only knows its own IP address and the ports on the server where the application runs.

Once communication with a server starts, the NAT device translates the private IP and port combination of the device connected on the private NAT interface to a temporary mapping of a public IP and port on the interface connected to the public network.

When the Consumer Network uses a private IP addressing schema (e.g. RFC 1918 [ADDR]) and the NAT device is port and/or address restricted, Consumer Network devices that receive incoming signalling (such as session setup, notification message, etc…) SHALL implement a mechanism to maintain open and active the necessary pin holes on the NAT device.

F.6.1 NAT Traversal for SIP based services

The two main NAT traversal scenarios are summarised below:

Hosted NAT for SIP over IPSec

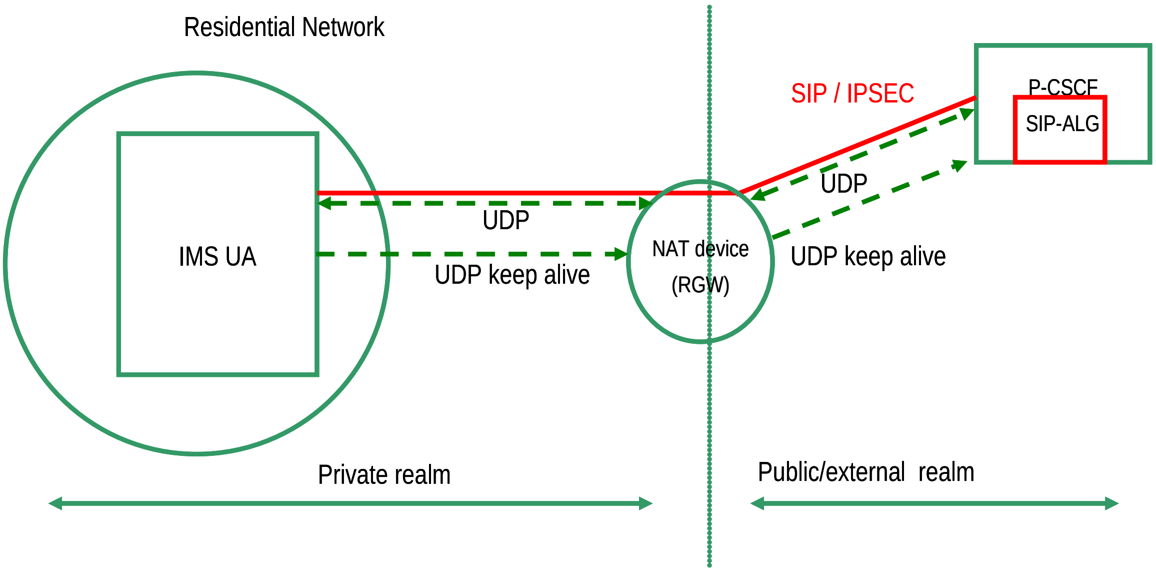
The NAT traversal solution defined for this scenario requires the following steps:

Step 1: Verify that the client (e.g. SIP UA) is behind a NAT device. In the IMS/3GPP scenario, this is achieved by using a plain text SIP message (the first SIP REGISTER). Note that within standard RFC IPSec the first step is performed directly within IKE (Internet Key Exchange), but within the IMS environment the authentication and key agreement phase is performed by using the AKA algorithm.

Step 2: The SIP UA establish the IPSec tunnel with the P-CSCF using the IETF IPSec NAT traversal solution that is based on UDP encapsulation;

Step 3: The UA maintains the pin holes open in the NAT device with UDP keep alive messages;

Step 4: All SIP message are sent over the IPSec tunnel (in both direction).



As there is a permanent communication path opened between the consumer device and the P-CSCF, it is always possible to send SIP messages between the entities involved in the communication (also when the SIP message request is originated from the network).

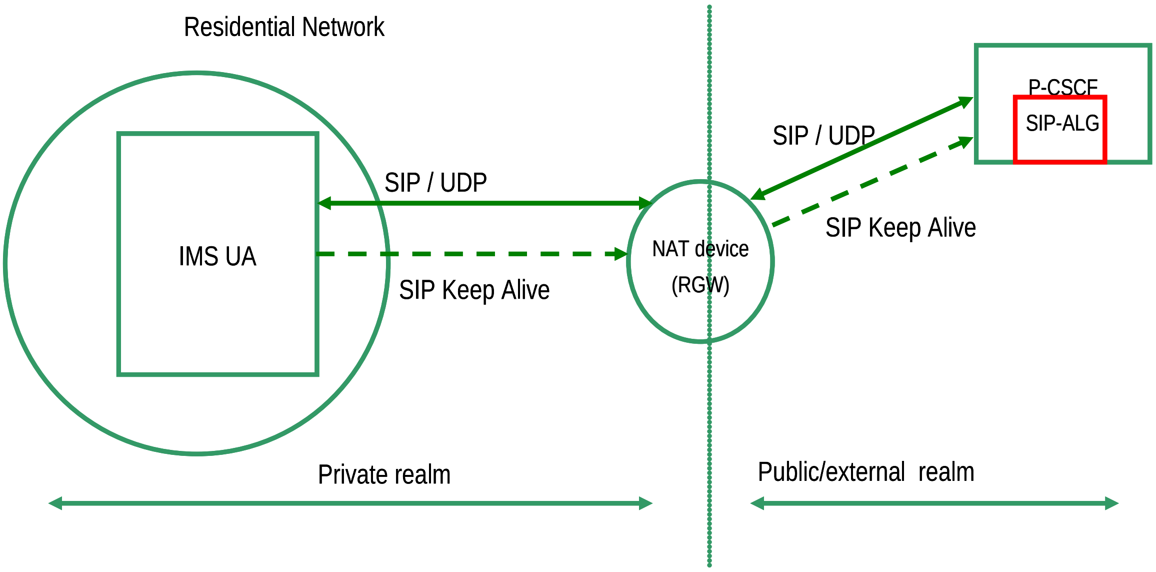
Hosted NAT for SIP plain text

The NAT traversal problem for SIP signalling can be solved by simply implementing keep alive messaging.

When it receives the first message, the P-CSCF can check the presence of a NAT device by comparing the address and port contained in the “Via” header to the actual IP and port combination in the received IP message.

Once registered with the SIP Registrar, the SIP UA MUST maintain the communication channel open by sending successive keep-alive packets before the binding expires in the NAT device.

The keep alive messages CAN either be SIP REGISTER or SIP OPTION sent by the client device.



As there is a permanent communication path opened between the consumer device and the P-CSCF, it is always possible to send SIP messages between the entities involved in the communication (also when the SIP message request is originated from the network).

It is REQUIRED to use symmetric signalling, this means that the proxy MUST send and receive data on the same port number.

F.6.2 NAT Traversal and keep-alive messages for unicast content streaming services

The NAT traversal problem for the media parameters transported within the SDP signalling can be solved by implementing a symmetric-RTP mechanism, as per RFC 4961 [RFC4961]:

When the OITF activates a unicast content streaming service it SHALL start to send keep-alive messages that consist of empty RTP packets with a payload type of 20 to the appropriate destination address and port, which depend on the scenario of the deployment and the model.

In addition to the NAT traversal problem, there is a NAT binding keep-alive problem. In general, it is not possible to determine or modify a retail NAT’s binding lifetime. Therefore, in order to keep the NAT bindings open, it is necessary to send keep-alives frequently. The REQUIRED frequency of the keep-alive messages is governed by a keep-alive timer. The value of the keep-alive timer SHALL be a random number between 24 and 29 seconds, if not configured. This timer MAY be configurable as described in RFC 4787 [RFC4787]. The empty RTP packet with a payload type of 20 is defined in TS 24.229 and endorsed in [TS124503].

These empty RTP packets with payload type 20 fulfill the following functions:

- The packets are used by the network for the discovery of the public client IP address and port (actually, the address and port of the WAN gateway) to use for the delivery of the RTP stream, and
- The packets are also used to keep the necessary pin holes on the NAT device open and active for the duration of the incoming RTP streaming.

This solution applies for all cases with the following difference:

IG and WAN GW in different physical devices

The functional entity that changes the destination address to the address and port discovered by the keep alive messages is the BGF (this is a component of the Transport Process Function defined in ES 282 003 [ES282003]), under the control of the P-CSCF (this is a component of the ASM defined in [OIPF_ARCH2]).

IG and WAN GW in one physical device

In this scenario the IG+WAN GW device SHALL catch and suppress the keep-alive messages.

WAN GW in one physical device (without IG)

The functional entity that changes the destination address to the address and port discovered by inspecting the keep-alive messages is the CDF, under the control of the CC. The keep-alive message, reaching the CDF, provides the client’s IP address and port to use for the delivery of the RTP stream.

Section 7.1.1.1, “RTSP Profile without SIP session management over UNIS-11 and NPI-10”, gives some recommendations for selecting a mechanism for keeping the RTSP session “alive.” In order to minimize OITF uplink traffic, the NAT binding keep-alives SHOULD be re-used as RTSP session keep-alive messages whenever possible. Note that it might be necessary for the OITF to send both NAT binding keep-alive messages and RTSP session keep-alives, for example, when the server cannot bind the RTP and RTSP sessions.

When the MPEG2TS is encapsulated in UDP (direct UDP), in order to keep the NAT bindings open the keep-alive SHALL be a UDP packet with the body of the packet filled with the value 20. The sender and destination IP addresses and ports SHALL follow the same rules as RTP keep-alives.

Annex G gives more detail and describes the informational flow for these cases.

Annex G System Infrastructure Mechanisms (informative)

G.1 NAT-T Informational flows for IPTV Services with SIP session management

The WAN Gateway itself can perform simple Network Address Translation (NAT) at the network and transport layer, but it is not able to modify the addresses embedded in the encapsulated signalling message. In order for the SIP services to work with NAT in this specification there are two possible alternatives that take into account the different deployment scenario defined in the architecture specification [OIPF_ARCH2]:

1. If the WAN Gateway and the IG are deployed together in a physical device the NAT-T can be solved with an embedded SIP application-level gateway (SIP-ALG). In this scenario the SIP signalling is generated from this device using the public address and the control of the incoming media streams can be performed internally by the device.

2. For other deployment scenarios, the NAT-T can be solved in the network with a SIP application-level gateway (SIP-ALG) in the P-CSCF that coordinate the work of the BGF; this solution is commonly defined Hosted-NAT

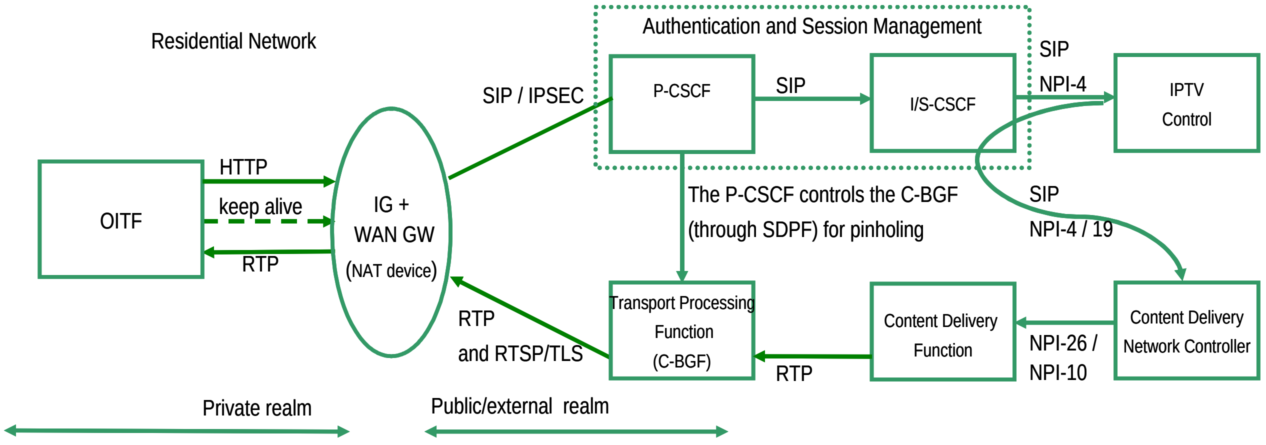
The main advantages for the Hosted-NAT architectures are:

- Minimal impact on the user device and the WAN gateway;
- Security protocols supported (e.g. IPSec)
- Main components already defined by TISPAN and 3GPP

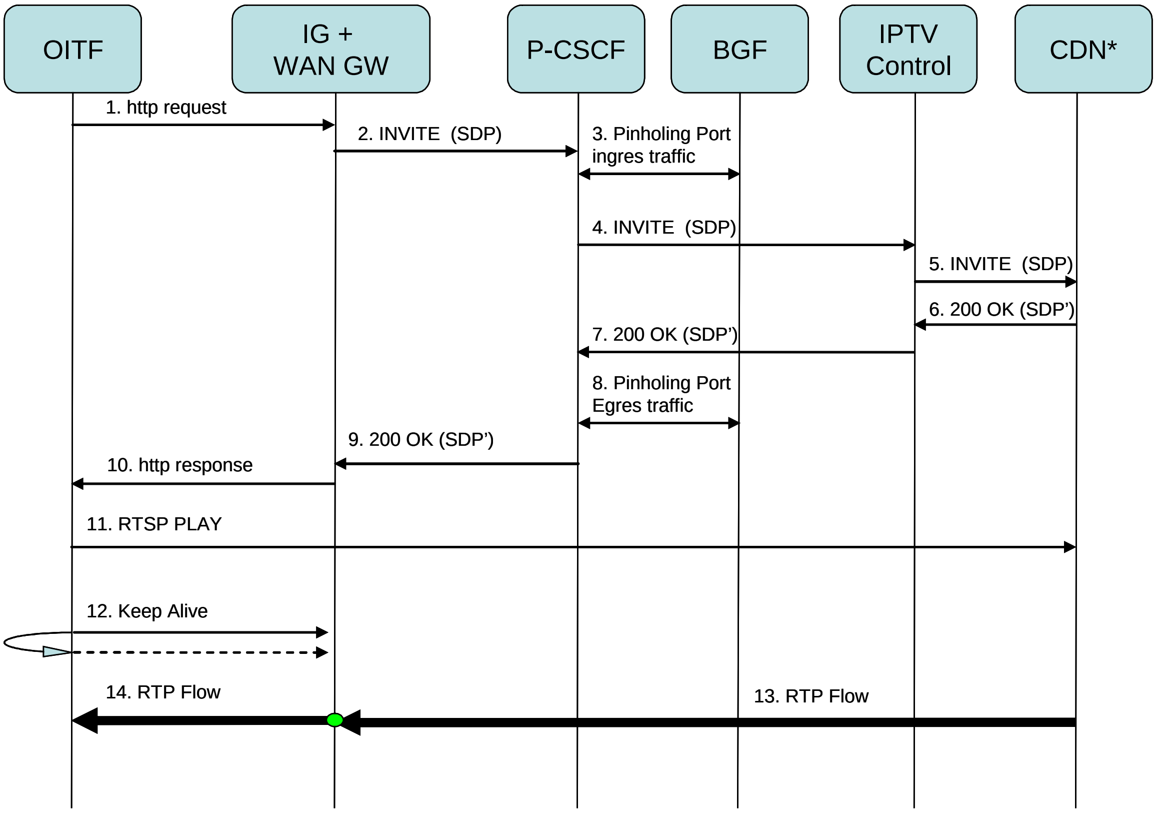
Since the Hosted-NAT solution is already used for other IMS services (e.g. voice call), it can be reused for the managed IPTV scenario in a general deployment option.

G.1.1 IG and WAN GW in one physical device

This section defines the NAT Traversal mechanism when the IG and WAN Gateway are deployed in the same device. This is considered to be a common scenario for managed networks relying on IMS. An embedded NAT-T solution and implemented internally in this device is considered an efficient mechanism.



The following informational flow describes the interaction between the functional entities for unicast content streaming services in this scenario, for simplicity the S-CSCF is not shown:



Step 1: The OITF sends a HTTP request for the desired unicast content streaming service to the IG (collocated with the WAN Gateway).

Step 2: The IG translates the request to a SIP INVITE with appropriate SDP description of the media request. The private address of the OITF is replaced with its public address.

Step 3: The P-CSCF (on Gq') requests to the BGF to open the pin hole for the ingress RTSP and eventually RTP media streams.

Step 4: The INVITE is forwarded to the IPTV Control.

Step 5: The INVITE is forwarded to the Cluster Delivery Network Controller (and Cluster Controller).

Steps 6-7: The 200 OK message is sent back to the P-CSCF.

Step 8: The P-CSCF on Gq' updates the allocation on the BGF for the RTSP and RTP media streams (egress traffic).

Step 9: The 200 OK message is sent back to the IG+WAN GW.

Step 10: The information carried with the 200 OK is sent to the OITF (inside the HTTP replay message).

Step 11: The OITF send a RTSP PLAY command to receive the media stream to the Content Delivery Function

Step 12: The OITF starts sending keep alive messages that consist of empty RTP packet with a payload type of 20 to the destination address and port contained in the 200 OK.

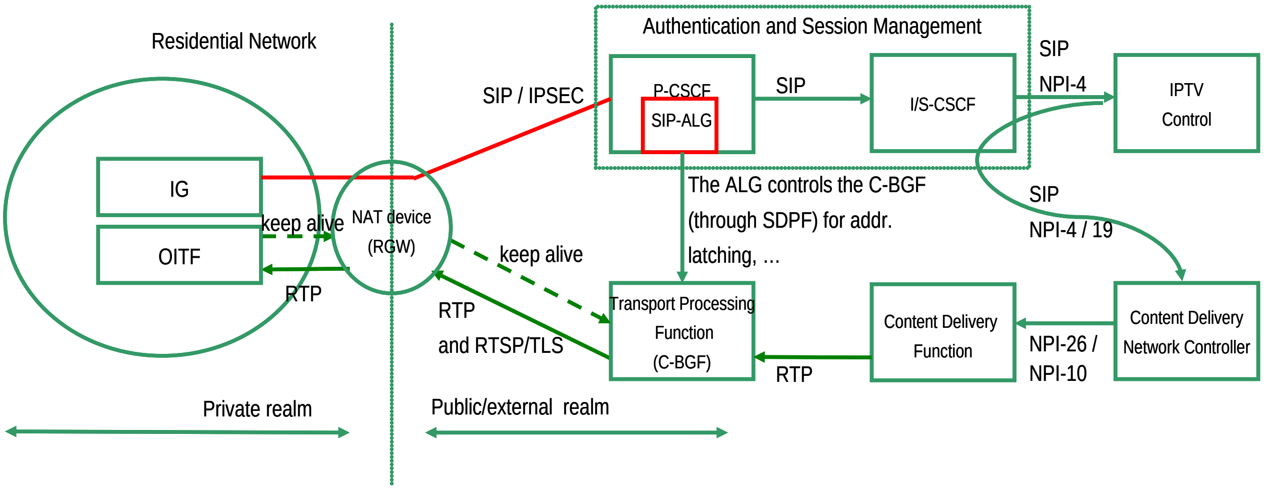
Note: In this scenario the IG+WAN GW device SHALL catch and suppress the keep alive messages, because the co-located ALG is aware of the NAT traversal mechanism, and SHAL be configured to suppress the keep alive messages.

Step 13: The CDN starts to send the media stream to the IG+WAN GW (by using the IP/Port received in the SDP packet as modified by the IG in step 2);

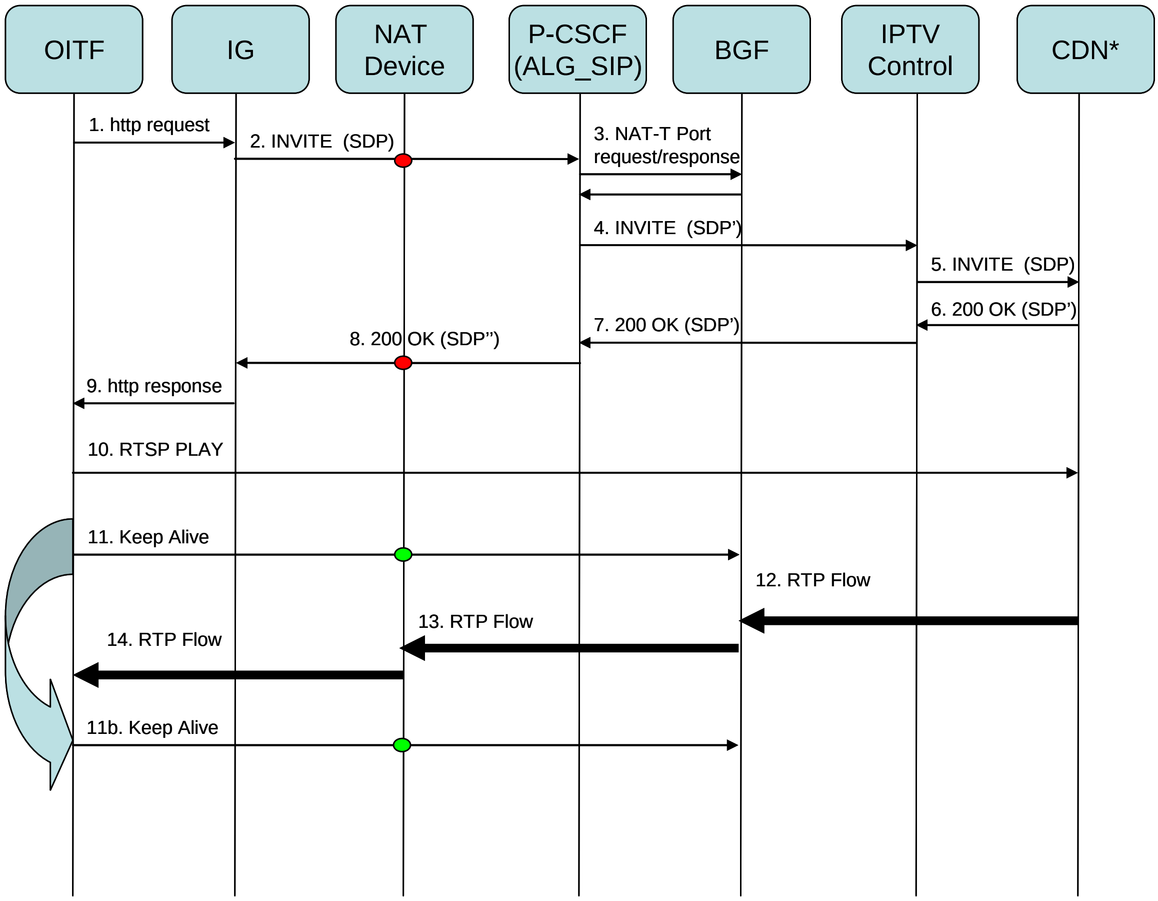
Step 14: The NAT device delivers the stream to the OITF by using its internal NAT table;

G.1.2 IG and WAN GW in different physical devices

In this scenario, the WAN GW is the NAT device and the solution is based on the 3GPP TS 33.203 [3G-SEC] IMS access NAT-T model.



It is REQUIRED to maintain a permanent communication path opened between the IG and the P-CSCF. This can be achieved using the hosted NAT solution described in Annex F.6.1. The following informational flow describes the interaction between the functional entities for unicast content streaming services and explains the need for the RTP keep-alive messages; for simplicity the S-CSCF is not shown:



Step 1: The OITF sends a HTTP request for a unicast content streaming service to the IG.

Step 2: The IG translates the request to a SIP INVITE with appropriate SDP description of the media requested.

Step 3: The P-CSCF over the Gq' interface requests the allocation of address ports on the BGF for the RTP media streams; this information is also used to update the IP and port address in the SDP message that describes the RTP stream.

Step 4: The INVITE with SDP updated is sent to the IPTV Control FE.

Step 5: The INVITE is forwarded to the Cluster Delivery Network Controller (and the Cluster Controller).

Steps 6-7: The 200 OK message is sent back to the P-CSCF.

Step 8: The SDP in the 200 OK answer is updated with the BGF address and port reserved for this media stream in the step 3.

Step 9: The information from the 200 OK is sent to the OITF.

Step 10: The OITF sends a RTSP PLAY command to receive the media stream from the Content Delivery Function.

Step 11: The OITF starts sending keep alive messages that consists of empty RTP packet with a payload type of 20 to the destination address and port contained in the 200 OK.

Step 12: The CDN starts to send the media stream to the BGF (by using the IP/Port received in the SDP packet).

Step 13: The BGF changes the destination address to the address and port discovered by the keep alive messages.

Step 14: The NAT device delivers the stream to the OITF by using its internal NAT table.

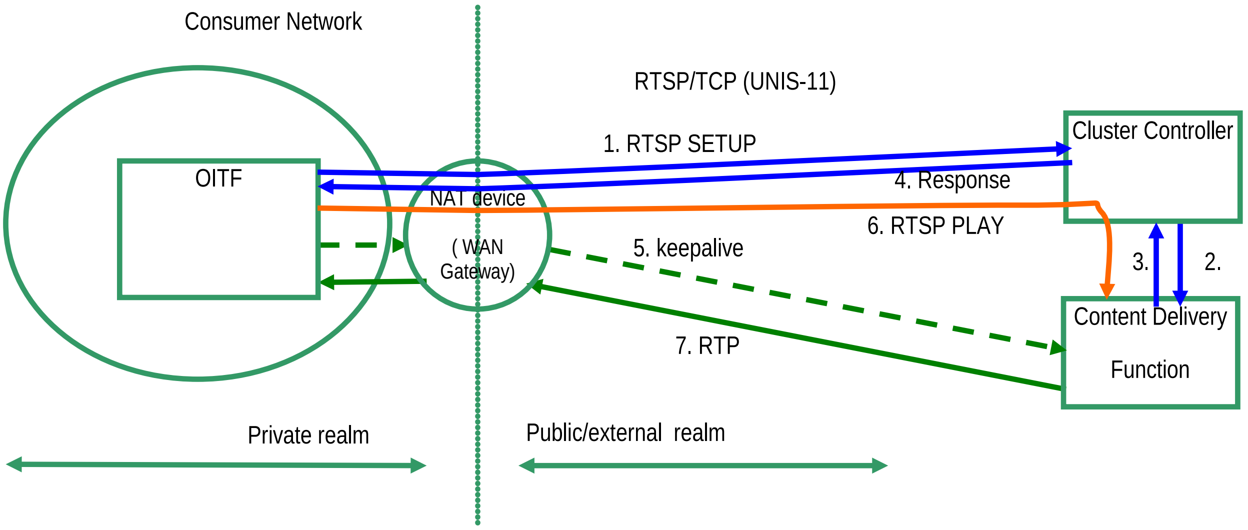
Note: The transport for RTSP is TCP only with either persistent or transient connection.

G.2 NAT-T Informational flows for IPTV Services

G.2.1 Symmetric-RTP

The Symmetric-RTP mechanism can also enable the NAT Traversal of RTP stream if the CDF for unicast content streaming service supports the detection of the external port number to send RTP stream towards the OITF. This solution will work even if multiple NAT devices exist between the CDF and the OITF.

The following flow describes the main steps involved in the Symmetric-RTP mechanism:



Step 1: The OITF sends an RTSP SETUP request to the CC (Cluster Controller). The CC detects the client's external IP address by the source IP address in the IP header.

Step 2: The CC forwards the information to the CDF. The CDF performs the same processing as the CC in step 1 to detect the presence of NAT. (The IP address declared in the SDP does not match the OITF address declared in the transport header of the RTSP SETUP message).

Step 3: The CDF returns the server IP address and port number of the RTP stream.

Step 4: The CC returns the response for SETUP request to OITF which contains the server IP address and port number of CDF.

Step 5: The OITF sends the keep-alive messages that consist of empty RTP packet with a payload type of 20 to the CDF. The keep-alive message punches a hole in the NAT device and then, reaching the CDF, provides the client's IP address and port to use for sending the RTP stream. The address latching is performed by the CDF if and only if the device is behind NAT as discovered during step 2.

Step 6: The OITF issues an RTSP PLAY request.

Step 7: The RTP packets can now be delivered from the CDF to the OITF.

G.3 Port mapping and NAT traversal for FCC/RET for multicast content services

The port mapping procedure defined in [PORTMAP] provides a means by which the OITF can signal to the FCC/RET server the port on which the FCC/RET server SHOULD transmit the FCC/RET RTP packets. However, the FCC/RET RTP packets transmitted by the FCC/RET server MAY not be able to reach the OITF when located behind a NAT. This is because the RAMS-R and NACK RTCP messages requesting the FCC/RET RTP packets, are sent by the OITF to the feedback target receive port on the FCC/RET server which is in general different from the port on which the FCC/RET RTP packets are sourced. I.e. the 4-tuple (UDP source and destination ports and IP source and destination addresses) of the FCC/RET request messages will not match with the 4-tuple of the FCC/RET packets, and hence MAY be blocked by a NAT positioned between the FCC/RET server and the OITF.

It MUST be noted that the main difference with previous sections in this Annex addressing NAT traversal for unicast content streams, is that there are no dedicated SIP and/or RTSP interactions between the OITF and the network, prior to the unicast (FCC/RET) RTP packet transmission.

The way NAT traversal is addressed for FCC/RET is dependent upon whether cookie signalling is used or not.

G.3.1 Cookie Signalling

In the port mapping process defined in [PORTMAP] and referred to in Annex M, "Fast Channel Change and Retransmission (FCC/RET)" the FCC/RET-enabled OITF sends a port mapping (request) RTCP message to the retransmission source entity RTP/RTCP receive port of the FCC/RET server (this is a single port as the FCC/RET server is REQUIRED to support RTP/RTCP port multiplexing for the unicast RTP FCC/RET session) prior to any RAMS (FCC) or NACK message exchange. The source port of this port mapping (request) message indicates to the FCC/RET server the (public) port on which the OITF wants to receive the FCC/RET packets. This information is encapsulated in a unique cookie that is generated by the FCC/RET server and which is transferred to the OITF along with the port mapping (response) message. This cookie is then sent by the OITF along with an FCC/RET request (RAMS-R or NACK FB RTCP message) to the Feedback Target entity of the FCC/RET server and hence provides the information *in-band* to the FCC/RET server on which port the FCC/RET RTP packets MUST be sent.

For FCC services, the RTCP port mapping request message will "prime" the NAT between the OITF and the FCC server for the short-lived and contiguous unicast flow of the FCC RTP/RTCP packets transmitted by the FCC server to the OITF. No keep-alive mechanism is REQUIRED for the NAT binding.

To keep the NAT binding open for RET services, similar to the symmetric RTP/UDP concept detailed in the previous sections of this annex, the OITF SHALL send "keep alive" messages to the retransmission source entity RTP/RTCP receive port of the RET server (which is different from the feedback target RTCP port of the FCC/RET server!). These "keep alive" messages can be empty RTP packets with payload type 20 and need to be sent with a frequency governed by an OITF keep-alive timer that SHALL be a random number between 24 and 29 seconds (as specified in Annex F.6.2, "NAT Traversal and keep-alive messages for unicast content streaming services").

Note: the initial RTCP port mapping request message can be considered the 1st "keep alive" packet

The process is depicted in Figure 18 in the context of RET service.

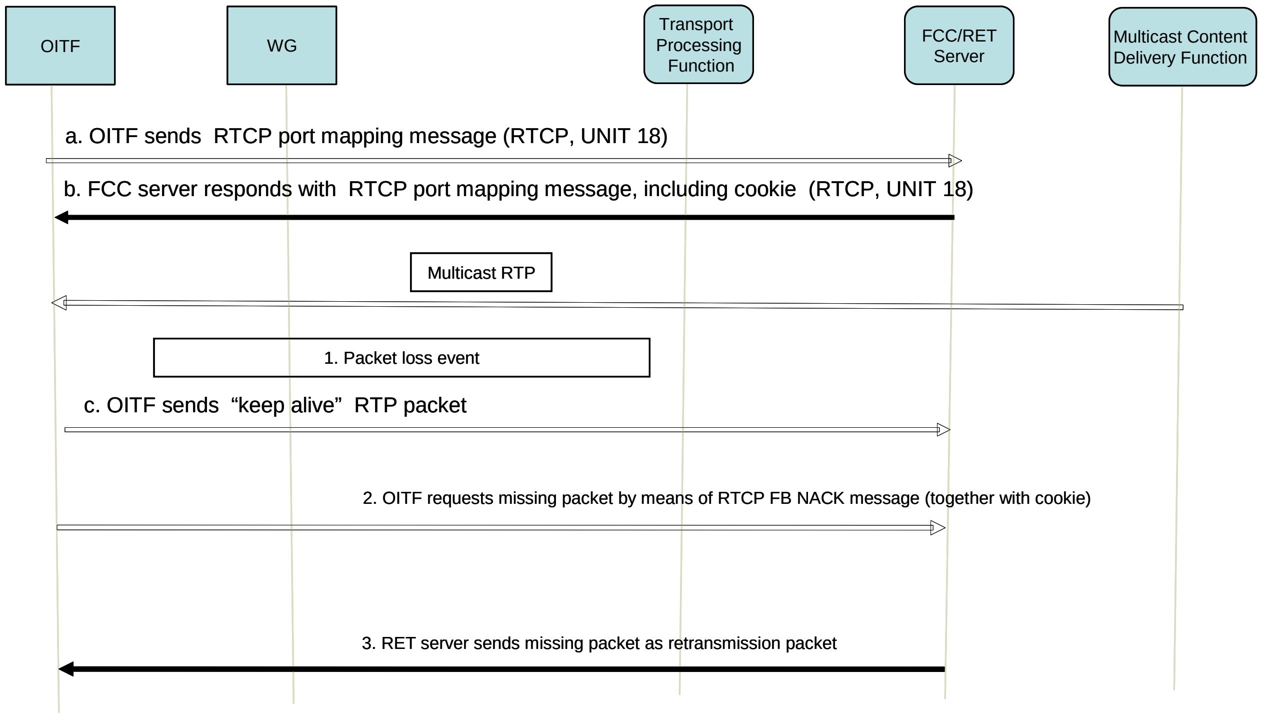

OITF
WG
Transport Processing Function
FCC/RET Server
Multicast Content Delivery Function
a. OITF sends RTCP port mapping message (RTCP, UNIT 18)
b. FCC server responds with RTCP port mapping message, including cookie (RTCP, UNIT 18)
Multicast RTP
1. Packet loss event
c. OITF sends "keep alive" RTP packet
2. OITF requests missing packet by means of RTCP FB NACK message (together with cookie)
3. RET server sends missing packet as retransmission packet

Figure 18: Port mapping process and keep-alive packets for RET

Figure 19 shows the relevant ports used for the unicast FCC/RET RTP and RTCP packet flows when the cookie signaling method is used.

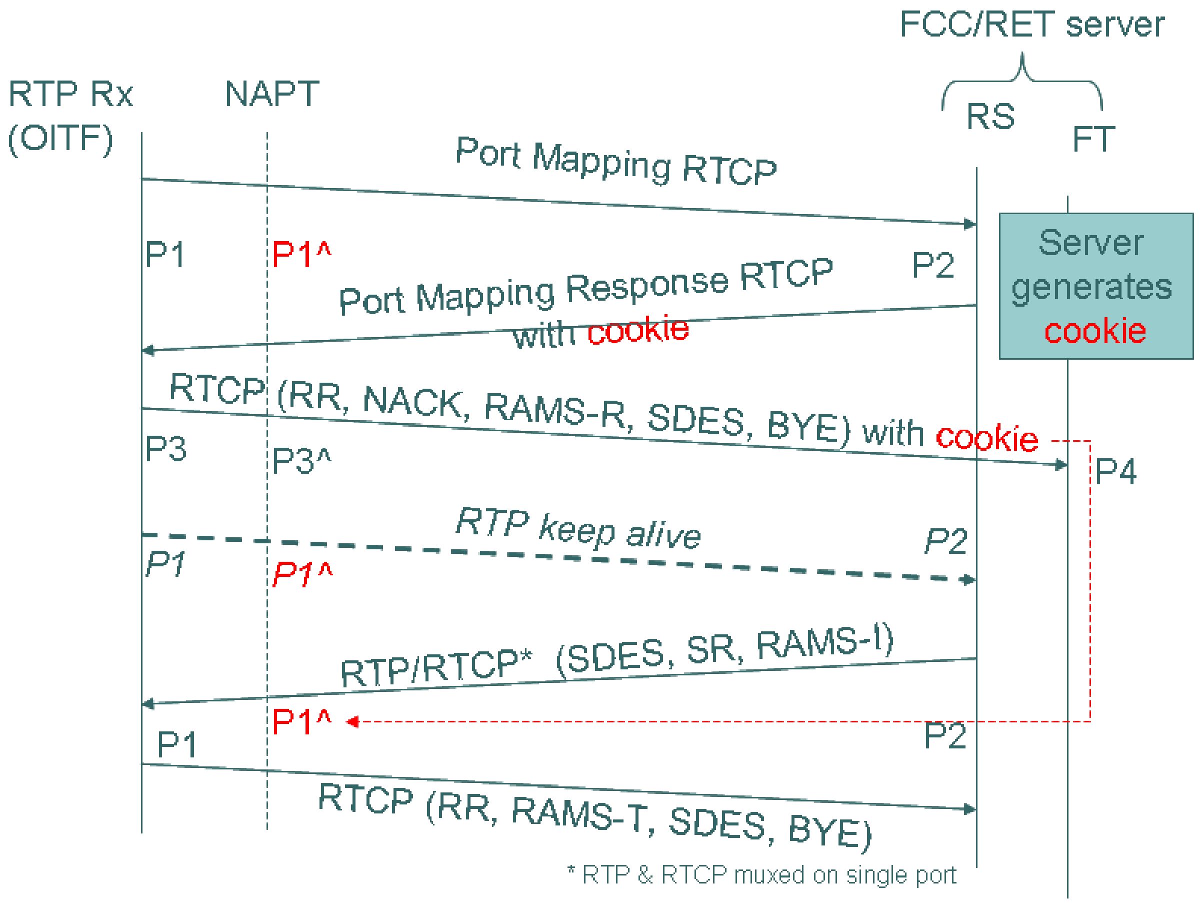

FCC/RET server
RTP Rx (OITF)
NAPT
RS
FT
Port Mapping RTCP
P1
P1^
P2
Server generates cookie
Port Mapping Response RTCP with cookie
RTCP (RR, NACK, RAMS-R, SDES, BYE) with cookie
P3
P3^
P4
RTP keep alive
P1
P1^
P2
RTP/RTCP* (SDES, SR, RAMS-I)
P1^
P1
P2
RTCP (RR, RAMS-T, SDES, BYE)
* RTP & RTCP muxed on single port

Figure 19: NAPT traversal for FCC/RET with cookie signalling

The OITF issues the RTCP port mapping message on Port P1 and are destined to port P2 of the retransmission server (RS) entity of the FCC/RET server. The NAPT will map the port P1 to Port P1 prime. The RS entity responds with a port mapping response message including the cookie using the same ports.

The RTCP messages sent by the OITF to the feedback target (FT) entity of the FCC/RET server (RAMS-R, NACK, RR, Bye combined with the Session Description Protocol Security Descriptions for Media Streams (SDES)) are sourced on port P3 and destined to port P4 (when passing the NAPT, the port P3 is mapped to a port P3 prime). NACKs or RAMS-R MUST be accompanied with the cookie. The cookie indicates to the FCC/RET server that the response to the NACK or RAMS-R MUST be sent to destination port P1 prime.

The RTP packets and associated RTCP packets (SR and RAMS-I combined with SDES) transmitted by the RS entity of the FCC/RET server to the OITF are sourced on port P2 and destined to port P1 prime. The NAPT will map this to the port P1 when forwarding the packet to the OITF.

In the case of RET services, the OITF sends a RTP keep-alive on source port P1 and destination port P2 to assure the NAT traversal of the RET packet(s). These RTP keep–alive messages can be ignored by the RET server.

The OITF sends RTCP packets (RR, RAMS-T, Bye combined with SDES) to the RS entity of the FCC/RET server with destination port P2 and with P1 as source port.

P4 and P2 are determined by SD&S, whereas the OITF determines the port P3 and P1.

G.3.2 Without Cookie Signalling

When the FCC/RET-enabled OITF is instructed to not use port mapping messages, i.e. no cookie signalling (see Annex M), the NAT traversal will be guaranteed because with this method the source port of the RAMS-R or NACK RTCP FB message indicates to the FCC/RET server the OITF receive port for FCC/RET RTP/RTCP packets and the FCC/RET server also assures that the source port of the FCC/RET packets matches the FCC/RET server FT receive port for the RTCP RAMS-R/ NACK messages. No keep-alive messages from the OITF are REQUIRED.

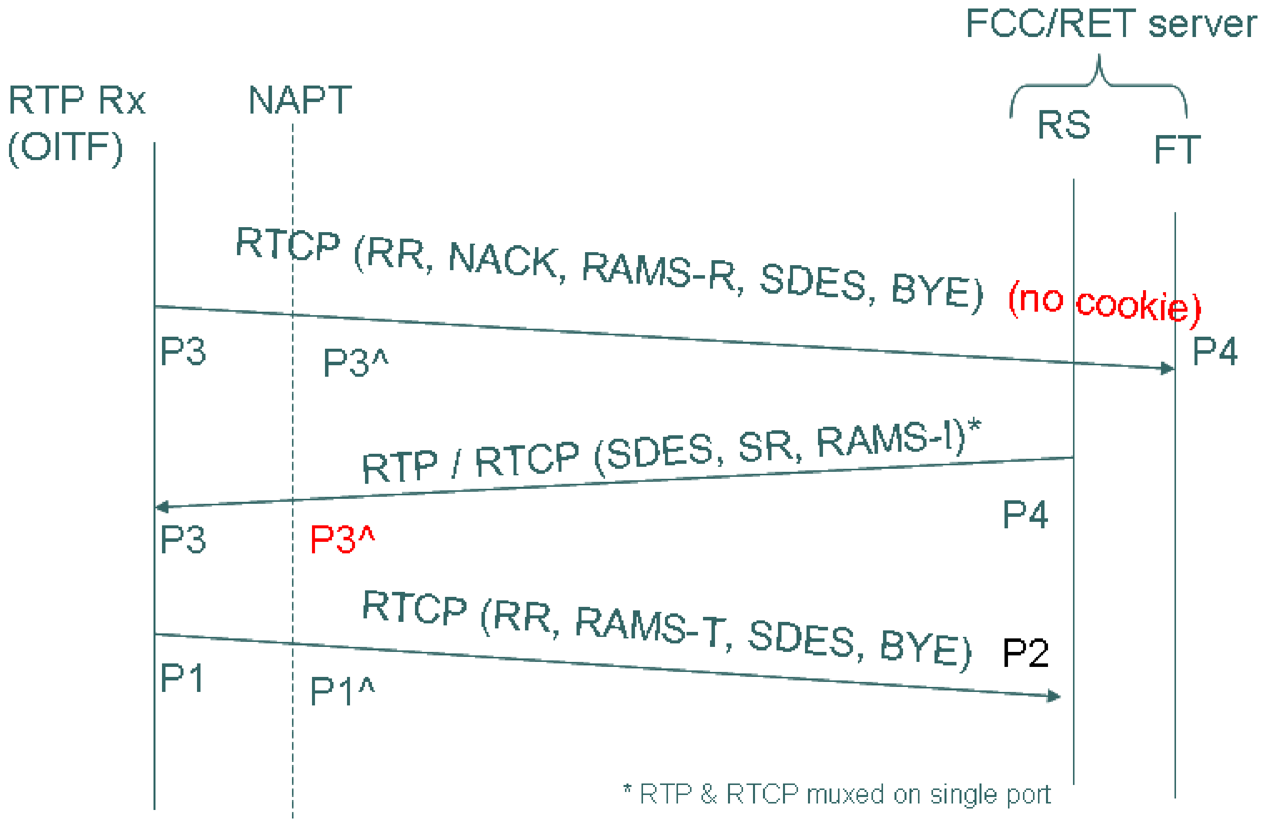

FCC/RET server
RTP Rx
(OITF)
NAPT
RS
FT
RTCP (RR, NACK, RAMS-R, SDES, BYE) (no cookie)
P3
P3^
P4
RTP / RTCP (SDES, SR, RAMS-I)*
P4
P3
P3^
RTCP (RR, RAMS-T, SDES, BYE)
P2
P1
P1^
* RTP & RTCP muxed on single port

Figure 20: NAPT traversal for FCC/RET without cookie signalling

Figure 20 shows the relevant ports used for the unicast FCC/RET RTP and RTCP packet flows in the non-cookie signaling method.

The RTCP messages sent by the OITF to the FT entity of the FCC/RET server (RAMS-R, NACK, RR, Bye combined with SDES) are sourced on port P3 and destined to port P4, but when passing the NAPT, the port P3 is mapped to a port P3 prime.

The RTP packets and associated RTCP packets (SR and RAMS-I combined with SDES) transmitted by the RS entity of the FCC/RET server to the OITF are sourced on port P4 and destined to port P3 prime. The NAPT will map this to the port P3 when FW-ing the packet to the OITF.

The OITF sends RTCP packets (RR, RAMS-T, Bye combined with SDES) to the RS entity of the FCC/RET server with destination port P2 and with P1 as source port.

P4 and P2 are determined by SD&S, whereas the OITF determines the port P3 and P1.

G.3.3 Correlation of RTCP messages sent to the RS and FT entity

In both the cookie and non-cookie signaling method, the RTCP messages sent by the OITF to the FT entity on Port P4 and to the RS entity on Port P2 are correlated by the FCC/RET server and linked to the same OITF by means of inspecting the SSRC identifier present in any RTCP packet and the unique OITF C-NAME present together with the SSRC in accompanying SDES RTCP packets.

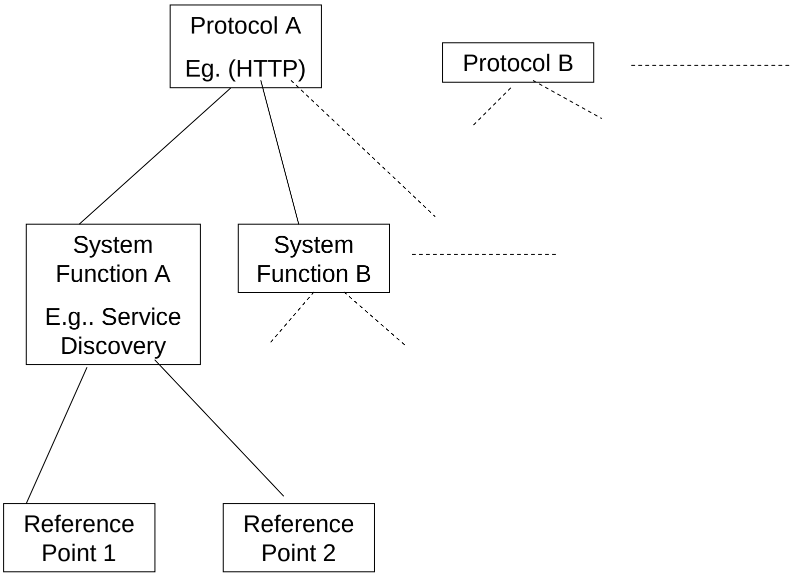
Each OITF has two SSRC identifiers that are associated with its unique C-NAME: one SSRC identifier used in the IP multicast RTP session (where the RTCP messages using that SSRC are directed to the FT entity of the FCC/RET server) and one SSRC identifier in the unicast session (where the RTCP messages using that SSRC are directed to the RS entity of the FCC/RET server). Those two SSRC-IDs MAY or MAY not have the same value, and hence the OITF CNAME SHALL be used by the FCC/RET server for RTCP message correlation purposes.

Annex H Presence XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:service:oitfpresence:2011"
  xmlns:tns="urn:oipf:service:oitfpresence:2011"
  xmlns:ct="urn:oipf:base:CommonTypes:2011"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="0.1">
  <xs:import namespace="urn:oipf:base:CommonTypes:2011"
    schemaLocation="base-CommonTypes.xsd" />
  <xs:element name="BroadcastTVService">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="BroadcastTV" type="tns:BroadcastTVPresenceType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="BroadcastTVPresenceType">
    <xs:sequence>
      <xs:element name="currentChannel" type="xs:string" />
      <xs:element name="currentProgram" type="xs:ct:ProgramIdType" />
      <xs:element name="serviceID" type="xs:string" />
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="Technology" type="tns:BroadcastTechnologyType" />
  </xs:complexType>
  <xs:simpleType name="BroadcastTechnologyType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DVB-T" />
      <xs:enumeration value="DVB-H" />
      <xs:enumeration value="DVB-S" />
      <xs:enumeration value="DVB-C" />
      <xs:enumeration value="DVB-T2" />
      <xs:enumeration value="DVB-S2" />
      <xs:enumeration value="DVB-C2" />
      <xs:enumeration value="ISDB-C" />
      <xs:enumeration value="ISDB-S" />
      <xs:enumeration value="ISDB-T" />
      <xs:enumeration value="ATSC-T" />
      <xs:enumeration value="ANALOG" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

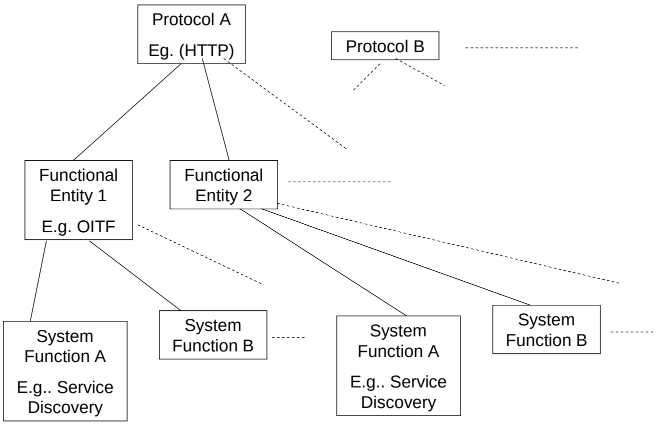
Annex I Protocol Procedure Section Structure (informative)

Each of the protocol sections of this document specifies the protocol procedures for a specific protocol (e.g. SIP, HTTP etc.). The sections have the following section and subsection structure.



This approach of structuring by protocol in the same way as the IMS IPTV Protocols specification developed in TISPAN SHOULD ensure that it is straightforward to investigate alignments with TISPAN. The structure chosen for this specification differs slightly from the TISPAN document structure, with the aim of helping the understanding of the end-to-end design, as it lends itself to the use of sequence charts to visualize the flow of a protocol through multiple components.

The actual TISPAN IMS IPTV Stage 3 specification structure is as follows:



Annex J OITF-specific TR-135 and TR-106 Remote Management Objects

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 with the same semantics (with a few exceptions) and the same types.

J.1 OITF-specific TR-135 Remote Management Object

The following table, obtained from "Table 1/TR-135 – Parameter list for an STB CPE device" in TR-135 [TR135], is the specific data model to manage an OITF.

Table 143: Parameter list for an OITF using TR-135

Parameter	R/W	Description
.STBService.{i}.Capabilities.		**The overall capabilities of the OITF. This is a constant read-only object, meaning that only a firmware update will cause these values to be altered**
MaxActiveAVStreams	R	max no of simultaneous AV streams active
.STBService.{i}.Capabilities.PVR.		**PVR Capability**
MaxIOStreams	R	0 means no PVR function, 1 mean PVR function (0,1)
.STBService.{i}.Capabilities.AudioDecoder.		**Audio decoder capabilities**
AudioStandards	R	Comma-separated list of audio standards supported by this OITF
.STBService.{i}.Capabilities.VideoDecoder.		**Video decoder capabilities**
VideoStandards	R	Comma-separated list of video standards supported by this OITF
.STBService.{i}.Capabilities.VideoDecoder.MPEG4Part10.		**Object describing the set of supported profiles and levels for this OITF. It also describes the set of audio standards supported when MPEG4 Part 10 is used as the video standard.**
AudioStandards	R	Comma-separated list of supported Audio Standards supported by the Player when associated with MPEG4 Part 10 video. Each item is taken from the list defined by .Capabilities.AudioDecoder.AudioStandards
ProfileLevelNumberOfEntries	R	Number of instances of ProfileLevel
.STBService.{i}.Capabilities.VideoDecoder.MPEG4Part10.ProfileLevel.{i}.		**Table to describe the set of profiles and levels combinations supported by the OITF when MPEG4 Part 10 is used as video standard. Each entry in this table refers to a distinct combination of profile and level. The table MUST include a distinct entry for each supported combination of these parameters.**
Profile	R	Comma-separated list of supported MPEG4 Part 10 profiles. Each item is an enumeration of: “BASELINE” “MAIN” “EXTENDED” “HIGH” “HIGH 10”

		“HIGH 4:2:2” “HIGH 4:4:4”
Level	R	Comma-separated list of supported MPEG4 Part 10 Levels. Each item is an enumeration of: “1” “1b” “1.1” “1.2” “1.3” “2” “2.1” “2.2” “3” “3.1” “3.2” “4” “4.1” “4.2” “5” “5.1”
.STBService.{i}.Capabilities.DRM.		**This object describes the characteristics of the Conditional Access and/or Digital Rights Management of the OITF.**
DRMSystems	R	Comma-separated list of unique identifiers of OIPF supported Content Protection systems Each item is an enumeration: “urn:dvb”casystemid:19188” “OIPF-DTCP-IP” “OIPF-CI+” “urn:dvb:casystemid:456 OIPF-CI+” “urn:dvb:casystemid:12345 OIPF-DCTP-IP"
.STBService.{i}.Capabilities.ServiceMonitoring.		**This object describes the capabilities of the ServiceMonitoring object.**
MaxActiveMainStreams	R	Maximum number of AV Main streams for which the STB can simultaneously collect statistics.
MinSampleInterval	R	Minimum sample interval in seconds that the STB MUST be able to support.
MaxReportSamples	R	Maximum number of samples of each statistic that the STB is able to store and report.
.STBService.{i}.Capabilities.FrontEnd.		**Front-end capabilities.**
.STBService.{i}.Capabilities.FrontEnd.DVBT.		**Capabilities of the DVB-T receiver.**
MaxActiveDVBTStreams	R	Maximum number of simultaneous active AV streams supported by the DVB-T FrontEnd. (0, 1)
.STBService.{i}.Capabilities.FrontEnd.IP.		**IP Front-End capabilities.**
MaxDejitteringBufferSize	R	Describes the maximum de-jittering buffer size, in bytes, supported by the OITF.
.STBService.{i}.Components.		**Details of OITF logical or physical internal components.**
FrontEndNumberOfEntries	R	Number of FrontEnd instances.

AudioDecoderNumberOfEntries	R	Number of AudioDecoder instances.
VideoDecoderNumberOfEntries	R	Number of VideoDecoder instances.
DRMNumberOfEntries	R	Number of DRM instances.
.STBService.{i}.Components.FrontEnd.{i}.		
Name	R	
.STBService.{i}.Components.FrontEnd.{i}.DVBT.		**DVB-T front-end details.**
.STBService.{i}.Components.FrontEnd.{i}.DVBT.Modulation.		**DVB-T modulation details.**
SNR	R	This parameter is normally the Signal/Noise ratio in the carrier band, measured in dB. In the context of OITF, this parameter (0 to 10) gives a signal quality value from 0 (no signal), 1 (weak signal), 5 (medium signal) to 10 strong signal, when information is available from the chipset
.STBService.{i}.Components.FrontEnd.{i}.IP.		**DVB-T front-end details.**
InboundNumberOfEntries	R	Number of Inbound instances.
.STBService.{i}.Components.FrontEnd.{i}.IP.RTCP.		**Parameters related to RTCP receiver report generation**
Enable	W	Enables or disables RTCP receiver report generation. If the OITF does not implement RTCP, then the OITF SHALL send error code 9001, “Request denied (no reason specified)” when the server tries to enable RTCP feedback.
Status	R	The status of RTCP receiver report generation. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition.
.STBService.{i}.Components.FrontEnd.{i}.IP.Inbound.{i}.		**Inbound IP streams currently entering the OITF via this front-end.**
SourceAddress	R	IP address of the source of the current stream content.
SourcePort	R	TCP or UDP port number of the source of the current stream content, or 0 if the content is not being delivered via IP or if not applicable.
URI	R	RFC 3986 URI that indicates the current source (possibly including Multicast group and port, if relevant) of the stream content, or an empty string if the source is not known or cannot be represented as a URI.
.STBService.{i}.Components.AudioDecoder.{i}.		**Audio decoder instance table. It contains data representing the current status of the Audio decoder.**
Status	R	The status of this audio decoder. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition.

Name	R	Human-readable name associated with this audio decoder.
AudioStandard	R	Audio standard currently being processed by this audio decoder, or an empty string if no audio standard is currently being processed.
.STBService.{i}.Components.VideoDecoder.{i}.		**Video decoder instance table. It contains data representing the current status of the video decoder.**
Status	R	The status of this video decoder. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition.
Name	R	Human-readable name associated with this video decoder.
MPEG4Part10	R	Path name of the MPEG4 Part 10 profile and level object instance.
ContentAspectRatio	R	Indicates the native aspect ratio of the content available at this decoder. Enumeration of: “4:3” “16:9”
.STBService.{i}.Components.DRM.{i}.		**This object describes the characteristics of the Digital Rights Management**
Status	R	The status of this DRM system. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition.
Name	R	Indicates a unique identifier for this DRM system. This name MUST appear in the .Capabilities.DRM.DRMSystems list.
.STBService.{i}.AVStreams.		**AV Streams object. If more than one AV stream can be active at a time, it MAY contain several AVStream instances.**
ActiveAVStreams	R	Number of AV streams currently active
AVStreamNumberOfEntries	R	Number of AVStream instances.
.STBService.{i}.AVStreams.AVStream.{i}.		**Details of each AVStream. AV streams are created statically. Each AV stream corresponds to a valid {FrontEnd, Audio- Decoder, VideoDecoder} instance combination**
Status	R	The status of this AV stream. Enumeration of: “Disabled” “Enabled” “Error_PVRWriteFailure” “Error_PVRReadFailure” “Error” Unspecified error (OPTIONAL) An AV stream is disabled if any of the referenced objects are disabled. If an AV stream is disabled then the values of other

		AV stream parameters are not significant. The “Error” value MAY be used by the CPE to indicate a locally defined error condition.
Name	R	Human-readable name associated with this stream
FrontEnd	R	Path name of the input FrontEnd object instance associated with this AV stream.
AudioDecoder	R	Path name of the Audio Decoder object instance associated with this AV stream.
VideoDecoder	R	Path name of the Video Decoder object instance associated with this AV stream.
Inbound	R	Path name of the inbound IP stream object instance associated with the FrontEnd for this AV stream.
.STBService.{i}.ServiceMonitoring.		**Contains statistics relating to the QoS / QoE of Main AV streams. Note that OITF devices do not support the collection of statistics while in STANDBY mode.**
SampleEnable	W	Enables or disables collection of Sample statistics.
SampleState	R	Indicates availability of Sample statistics. Enumeration of: "Disabled" Collection is disabled "Enabled" Collection is enabled "Trigger" Collection is enabled and the ACS SHOULD now fetch the collected data The transition from Enabled -> Trigger -> Enabled MUST be instantaneous and so will result in only a single value change for notification purposes.
SampleInterval	W	The sample interval in seconds.
ReportSamples	W	The number of samples that the OITF can store and report for each statistic.
TimeReference	W	An absolute time reference in UTC to determine when sample intervals will complete.
ReportStartTime	R	The absolute time at which the sample interval for the first stored sample (for each statistic) started.
ReportEndTime	R	The absolute time at which the sample interval for the last stored sample (for each statistic) ended.
MainStreamNumberOfEntries	R	Number of MainStream instances.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.		**List of Main AV stream objects. Each instance is associated with a specified service type and will collect statistics only for the main stream that matches that service type.**
Enable	W	Enables or disables collection of Total and Sample statistics for this object instance.
Status	R	Total and Sample statistics collection status for this object instance. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition.
ServiceType	W	Service type associated with this main stream instance, or an empty string if this instance is disabled. ServiceType is taken from the list: “IPTV”

		“VoD” “IP” “CAB” “DTT” “SAT” “PVR”
AVStream	R	Path name of the Main AV stream object instance currently associated with this ServiceMonitoring main stream instance.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.		**Total statistics since this ServiceMonitoring main stream instance was last enabled or Total statistics were last reset.**
Reset	W	When set to true, resets Total statistics for this ServiceMonitoring main stream instance. Setting it to false has no effect. The value is not saved in device state and is always false when read.
ResetTime	R	Number of seconds since the Total statistics were last enabled or reset for this ServiceMonitoring main stream instance.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.DejitteringStats.		**Total de-jittering statistics for this ServiceMonitoring main stream instance.**
Overruns	R	Total number of times the receive jitter buffer has overrun for this AV stream.
Underruns	R	Total number of times the receive jitter buffer has underrun for this AV stream.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.RTPStats.		**Total RTP statistics for this ServiceMonitoring main stream instance.**
PacketsReceivedBeforeEC	R	Total number of RTP packets received for this AV stream. These statistics are collected before any EC, if available, is applied.
PacketsLostBeforeEC	R	Total number of RTP packets lost for this stream. These statistics are collected before any EC, if available, is applied.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.MPEG2TSStats.		**Total MPEG2-TS statistics for this ServiceMonitoring main stream instance.**
TSPacketsReceived	R	Total number of MPEG2-TS packets received for this AV stream.
PacketDiscontinuityCounter	R	Total number of MPEG2-TS Discontinuity errors that have been captured for this AV stream. This parameter accumulates all of the discontinuities observed for all currently monitored PIDs.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.VideoDecoderStats.		**Total video decoder application layer statistics for this ServiceMonitoring main stream instance.**
ILostFrames	R	The number of I frames that could not be reproduced by the OITF for this AV stream.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.RTPStats.		**RTP Sample statistics for this Service-Monitoring main stream instance.**
SampleSeconds	R	Comma-separated list; each entry is the number of seconds during which RTP data was collected for this AV stream during the sample interval.
PacketsExpected	R	Comma-separated list; each entry is the total

		number of RTP packets expected for this AV stream during the sample interval
PacketsLostBeforeEC	R	Comma-separated list; each entry is the total number of RTP packets lost for this AV stream during the sample interval.
PacketsReceivedBeforeEC	R	Total number of RTP packets received for this AV stream. These statistics are collected before any EC, if available, is applied.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.MPEG2TSStats.		**MPEG2-TS Sample statistics for this Service-Monitoring main stream instance.**
SampleSeconds	R	Comma-separated list; each entry is the number of seconds during which MPEG2-TS data was collected for this AV stream during the sample interval.
TSPacketsReceived	R	Comma-separated list; each entry is the total number of MPEG2-TS packets received for this AV stream during the sample interval.
PacketDiscontinuityCounter	R	Comma-separated list; each entry is the total number of MPEG2-TS Discontinuity errors that were captured for this AV stream during the sample interval.
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.DejitteringStats.		**De-jittering Sample statistics for this ServiceMonitoring main stream instance.**
SampleSeconds	R	Comma-separated list; each entry is the number of seconds during which de-jittering data was collected for this AV stream during the sample interval.
Overruns	R	Comma-separated list; each entry is the total number of times the receive jitter buffer has overrun for this AV stream during the sample interval.
Underruns	R	Comma-separated list; each entry is the total number of times the receive jitter buffer has underrun for this AV stream during the sample interval.

J.2 OITF-specific TR-106 Remote Management Object

The following table, obtained from “Table 3 – Common Object definitions for Device:1” in TR-106 Amendment 1, “Data Model Template for TR-069-Enabled Devices” [TR106], is the specific data model to manage an OITF..

Note that:

For Device.DeviceInfo, the 3 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 section 6.1.3.2.1, “User Identity Modelling”):

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).

All Device.LAN parameters are read-only

Table 144: Parameter list for an OITF using TR-106

Parameter	R/W	Description
Device.		
DeviceSummary	R	The DeviceSummary parameter is defined to provide an explicit summary of the top-level data model of the device, including version and profile information.
Device.DeviceInfo.		**General information about the device, including its identity and version information.**
Manufacturer	R	The manufacturer of the CPE (human readable string).
ManufacturerOUI	R	In the context of OIPF, this parameter is the hexadecimal value of the first 3 bytes of SHA-1(X)
ModelName	R	Model name of the CPE (human readable string).
Description	R	A full description of the CPE device (human readable string).
ProductClass	R	In the context of OIPF, this parameter is always "OIPF"
SerialNumber	R	In the context of OIPF, this parameter is the hexadecimal value of the remaining bytes (from 4th on) of SHA-1(X)
SoftwareVersion	R	A string identifying the software version currently installed in the CPE.
Device.ManagementServer.		**This object contains parameters relating to the CPE’s association with an ACS.**
URL	W	URL for the CPE to connect to the ACS using the CPE WAN Management Protocol. This parameter MUST be in the form of a valid HTTP or HTTPS URL. The “host” portion of this URL is used by the CPE for validating the ACS certificate when using SSL or TLS. Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory value. If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the situation that the original value is restored as the result of a factory reset.
Username	W	Username used to authenticate the CPE when making a connection to the ACS using the CPE WAN Management Protocol. This username is used only for HTTP-based authentication of the CPE. Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory value. If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the situation that the original value is restored as theresult of a factory reset.
Password	W	Password used to authenticate the CPE when making a connection to the ACS using the CPE WAN Management Protocol. This password is used only for HTTP-based authentication of the CPE. When read, this parameter returns an empty string, regardless of the actual value. Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory value. If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the situation that the original value is restored as the result of a factory reset.
PeriodicInformEnable	W	Whether or not the CPE MUST periodically send CPE information to the ACS using the Inform method call.
PeriodicInformInterval	W	The duration in seconds of the interval for which the CPE MUST attempt to connect with the ACS and call the Inform method if PeriodicInformEnable is true.
PeriodicInformTime	W	An absolute time reference in UTC to determine when the CPE SHOULD initiate the Inform method calls. Each Inform call MUST occur at this reference time plus or minus an integer

		multiple of the {{param|PeriodicInformInterval}}. A zero dateTime value (0000-00-00T00:00:00) indicates that no particular time reference is specified. That is, the CPE MAY locally choose the time reference, REQUIRED only to adhere to the specified
ParameterKey	R	ParameterKey provides the ACS a reliable and extensible means to track changes made by the ACS. The value of ParameterKey MUST be equal to the value of the ParameterKey argument from the most recent successful SetParameterValues, AddObject, or DeleteObject method call from the ACS.
ConnectionRequestURL	R	HTTP URL for an ACS to make a Connection Request notification to the CPE.
ConnectionRequestUsername	W	Username used to authenticate an ACS making a Connection Request to the CPE.
ConnectionRequestPassword	W	Password used to authenticate an ACS making a Connection Request to the CPE. When read, this parameter returns an empty string, regardless of the actual value.
Device.GatewayInfo.		**This object contains information associated with a connected Internet Gateway Device.**
ManufacturerOUI	R	Organizationally unique identifier of the associated Internet Gateway Device. An empty string indicates that there is no associated Internet Gateway Device that has been detected.
ProductClass	R	Identifier of the product class of the associated Internet Gateway Device. An empty string indicates either that there is no associated Internet Gateway Device that has been detected, or the Internet Gateway Device does not support the use of the product-class parameter.
SerialNumber	R	Serial number of the associated Internet Gateway Device. An empty string indicates that there is no associated Internet Gateway Device that has been detected.
Device.LAN.		**This object contains parameters relating to IPbased LAN connectivity of a device.**
AddressingType	R	The method used to assign an address to this interface. Enumeration of: “DHCP” “Static”
IPAddress	R	The current IP address assigned to this interface.
SubnetMask	R	The current subnet mask.
DefaultGateway	R	The IP address of the current default gateway for this interface.
DNSServers	R	Comma-separated list of IP address of the DNS servers for this interface.

Annex K New Event package for SIP SUBSCRIBE /NOTIFY (informative)

Event: oipf-spdlist;DomainName="koreatelecom.co.kr".

The newly created event name SHALL be "oipf-spdlist"

- Also, the *DomainName* parameter can have either "*ALL*" or a certain service provider's domain name. If the value of *DomainName* parameter SHALL be "*ALL*", IPTV SP Discovery FE SHOULD return all SP discovery information of all service providers. However, if the *DomainName* value SHALL be a specific domain address, the SP Discovery FE SHOULD return the request specific SP discovery information.

Extending the existing Accept-Encoding:

- The value of "Accept-Encoding" can be either "gzip" or "xml-oipf-bim". When OITF requests with "xml-oipf-bim", the SP Discovery FE SHOULD return the SP Discovery XML document which SHALL be encoded with BiM.

New Event package for SIP NOTIFY request

- The Event header of the SIP NOTIFY request SHALL be set with "oipf-spdlist".
- Event: oipf-spdlist;
- The Content-Encoding SHALL be either "gzip" or "xml-oipf-bim".
- The "effective-by" parameter for the event header SHALL be set to 0.
- The Content-Type SHALL be "application/vnd.oipf.spdlist+xml".

Annex L Overview of Notification Services in OIPF R2 (informative)

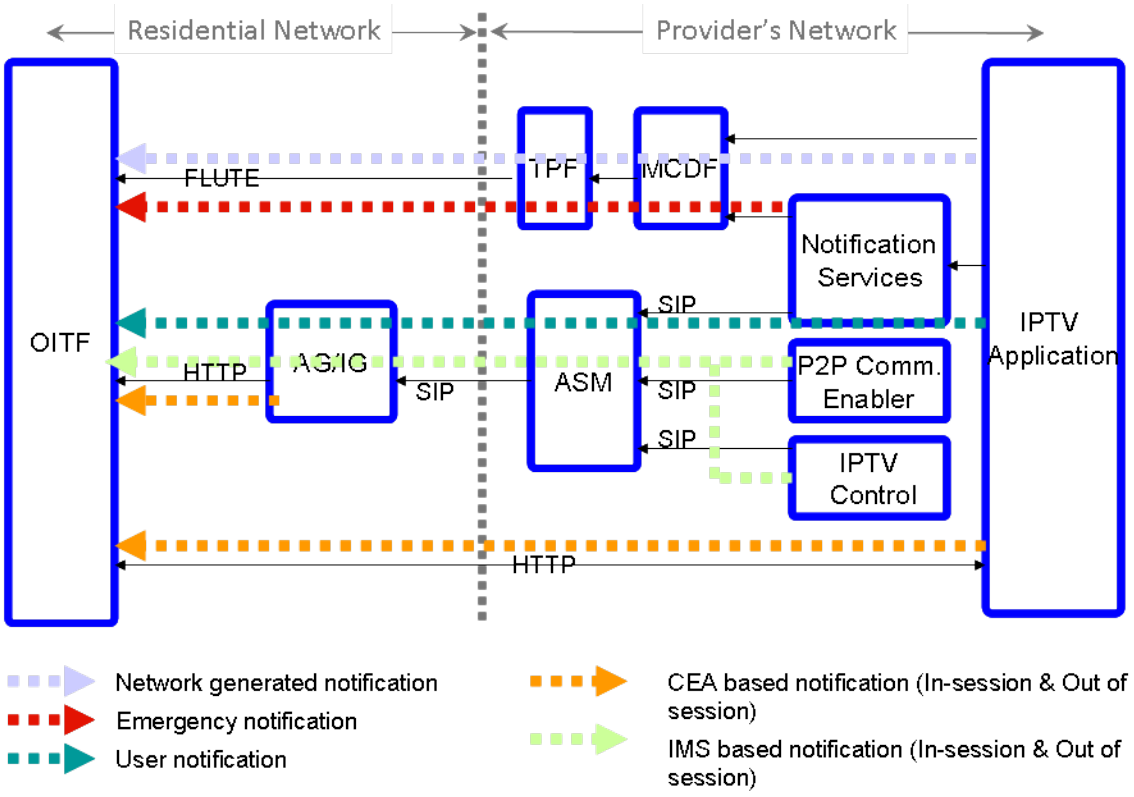


Figure 21: Overview of notification services

Several notification mechanisms are defined for different purpose. The above picture gives an overview of them.

- Network generated notification: defined in present document. In this mechanism, notification is transmitted via multicast session and therefore particularly suitable for service provider to provide notification services to a number of users in the same time. The typical service is notification associated with a multicast scheduled content service.
- Emergency notification: defined in present document. Similar with network generated notification mechanism, the notification is also transmitted via multicast session. This mechanism is used for service provider to provide emergency notification services.
  User notification: defined in present document. In this mechanism, notification is subscription based and is transmitted via SIP and proxied by IG. The typical case is for service provider to provide the reminder service about program starts.
- CEA based notification (In-session & 3rd party): defined in section 5.3.1 in [OIPF_DAE2] based on [CEA-2014-A]. In this mechanism, notification can be transmitted either via multicast session within home network or via HTTP Polling (within or outside home network).
- IMS event notification (In-session & 3rd party): defined in section 5.3.2 in [OIPF_DAE2]. In this mechanism, notification can be transmitted either via SIP and proxied by IG. The typical case is providing Call-ID, Messaging, Chatting service.

Following is the protocol and delivery manner for various notification mechanisms:

Table 145: Summary of notification mechanisms

	Unicast/multicast	**Protocol**
NG Notification	Multicast	FLUTE
Emergency Notification	Multicast	FLUTE
User Notification	Unicast	SIP + HTTP
CEA-based Notification	Multicast*/Unicast	HTTP/UDP
IMS event Notification	Unicast	SIP + HTTP
* Here multicast is only applicable within the home network domain.		

Annex M Fast Channel Change and Retransmission (FCC/RET)

M.1 Application Layer Retransmission (RET)

Retransmission service is based on a server-based technique to prevent visual and audio distortions for IPTV end users, caused by the loss of packets that MAY occur in the service provider network or home network.

RET MAY be applied to both multicact and unicast content services carried over RTP.

M.1.1 Unicast RET for multicast content service

The OITF interacts with a RET server to make use of the unicast RET service for multicast content service. This RET server receives and caches the IP multicast streams transporting the content services for a limited time, and acts simultaneously as

- the unicast feedback target (FT) entity for the RTP IP unicast session transporting the content service, where the OITF sends its RTCP messages -including the Retransmission requests – in unicast to this FT.
- retransmission source (RS) transmitting RET packets from its cache in a dedicated unicast RTP session.

It MUST be noted that when a multicast content service is both RET-enabled and FCC-enabled, the RET server and FCC server SHALL be one and the same.

The following steps take place for the unicast RET service when a packet loss event occurs in the network impacting the OITF:

Step 1: The OITF detects one or several missing RTP packets based on RTP sequence number tracking.

Step 2: The OITF requests the retransmission of any missing packets with a RTCP FB NACK message to the FT entity of the RET server.

Step 3: The retransmission source entity of the RET server sends the missing packet(s) as RTP retransmission packet(s), formatted as defined in [RFC4588].

The received retransmitted packet(s) is/are put in the original sequence slot in the OITF buffer, ready to be decoded.

- Cookie signalling

  a) The OITF sends a port mapping RTCP (request) message to the RET server. The source port of this message indicates to the RET server the OITF receive port to be used for RET RTP packets.

  b) The FCC/RET server responds with a RTCP port mapping (response) message containing a cookie, which SHALL be sent by the OITF in a dedicated RTCP message together with any subsequent NACK RTCP message requesting RET.

  The RET server processes the cookie received with every RTCP NACK from the OITF, and this way knows the port for the subsequent RET packet to be sent to that OITF.

  This RTCP port mapping message exchange needs to take place prior to the requesting and delivery of RET packets, and SHALL be performed each time the OITF connects to a new IP multicast content service where the cookie remains valid until the OITF disconnects from the IP multicast stream. If an OITF makes use of both the RET and FCC services for the multicast content service, there is a single port mapping message exchange process, as FCC and RET packets are transmitted in the same RTP session. Hence the same cookie is transmitted along with the FCC request and RET request (NACK) RTCP messages.

- Without cookie signalling

  With this method the source port of the NACK RTCP message indicates to the RET server the OITF receive port for RET RTP packets. The RET server SHALL also assure that the source port of the RET packet matches the RET server receive port for the RTCP NACK messages.

Note 1: A RET-enabled OITF SHALL support RTP/RTCP muxing, i.e. the OITF SHALL be able to receive the RTP and RTCP packets in the unicast RET session on the same receive port, regardless if method a or b is used by the OITF.

Note 2: For a RET-enabled multicast content service where no cookie signaling is used, the unicast RET packets will traverse any NAT between OITF and RET server because the NAT is primed by the RTCP FB NACK message. Hence, the OITF does not need to send RTP Keep-alive messages When the cookie signaling method is used, the OITF SHALL need to send RTP keep-alives to guarantee the NAT traversal (see Annex G.3, “Port mapping and NAT traversal for FCC/RET for multicast content services”) for the RET packets.

SD&S signals whether the port mapping messaging process (cookie signaling method) MUST be performed or not by the OITF for the RET/FCC-enabled multicact content service. The cookie signaling method SHALL be supported by the OITF.

Figure 22 shows schematically the steps involved when an OITF makes use of the unicast RET service for multicact content service. Steps a and b are only present when the OITF uses the cookie signalling method.

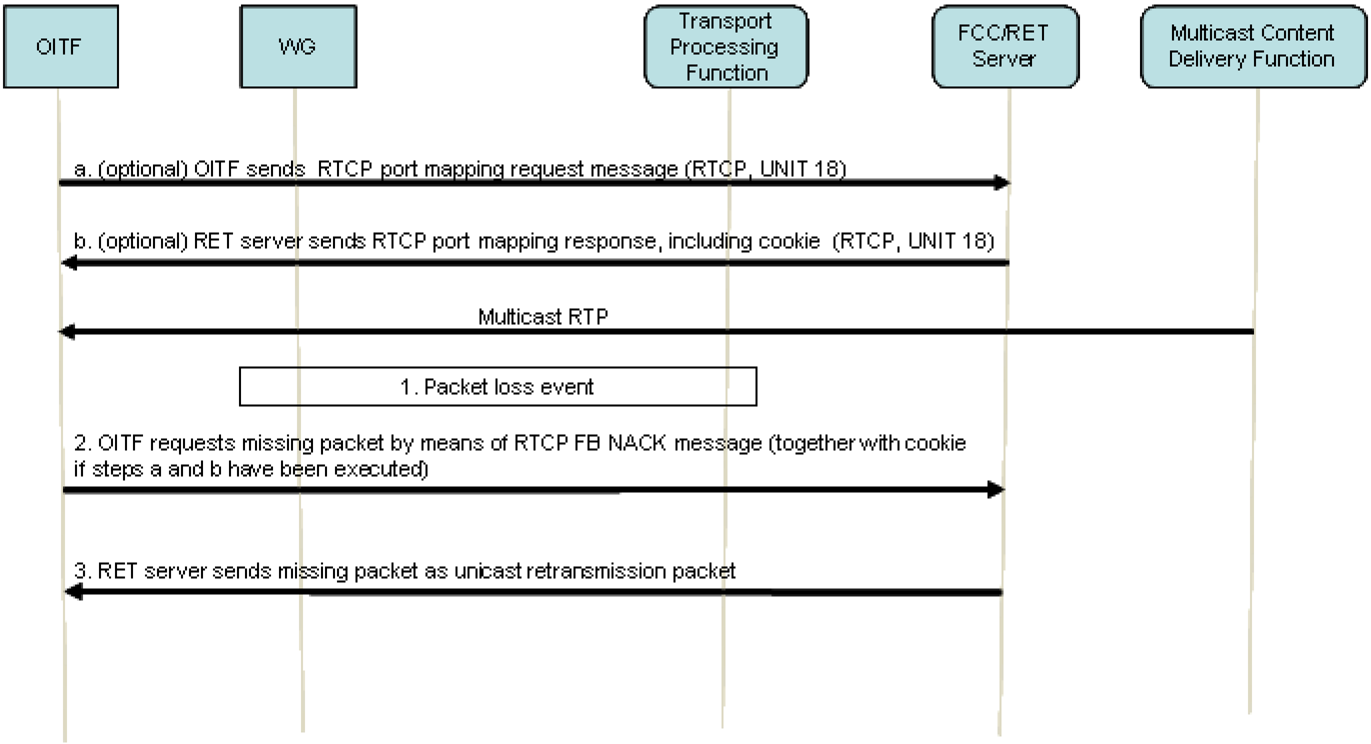

OITF
WG
Transport Processing Function
FCC/RET Server
Multicast Content Delivery Function
a. (optional) OITF sends RTCP port mapping request message (RTCP, UNIT 18)
b. (optional) RET server sends RTCP port mapping response, including cookie (RTCP, UNIT 18)
Multicast RTP
1. Packet loss event
2. OITF requests missing packet by means of RTCP FB NACK message (together with cookie if steps a and b have been executed)
3. RET server sends missing packet as unicast retransmission packet

Figure 22: Call flow for multicast content service with unicast retransmission service

Note: For multicast content services, RET MAY also be offered as a multicast service (See Annex M.1.3, “Multicast RET for multicast content service”.).

M.1.2 Unicast RET for unicast content service

The same steps 1, 2 and 3 take place as described for the unicast RET-enabled multicst content service (see Annex M.1.1, “Unicast RET for multicast content service”), but the server is the RET-server function that is embedded in the unicast media streaming server.

M.1.3 Multicast RET for multicast content service

Multicast RET for multicast content service enables scalable packet loss recovery for packet loss events in the multicast stream taking place deep in the network. Multicast RET service SHALL be supported by the OITF when the OITF supports Unicast RET service for multicact content service.

Similar to unicast RET, the IPTV service provider needs to assure there is extra bandwidth available –as compared to a multicact content service without RET- for this RET service. The extra bandwidth budget can be shared among unicast and multicast RET for multicact content service.

When multicact RET service is offered, along with the joining the IP Multicast on which the content service is transported, an OITF supporting RET SHALL also join the IP multicact RET sourced by the RET server, on which the OITF MAY receive RTCP FB NACKs and RET packets, triggered by packet loss events deep in the network.

A distinction can be made whether the packet loss event takes place upstream of the RET server or downstream of the RET server. Figure 23 shows a topology example illustrating where packet losses in the IP multicact content service tree MAY occur relative to the RET server and to the OITFs serviced by this RET server.

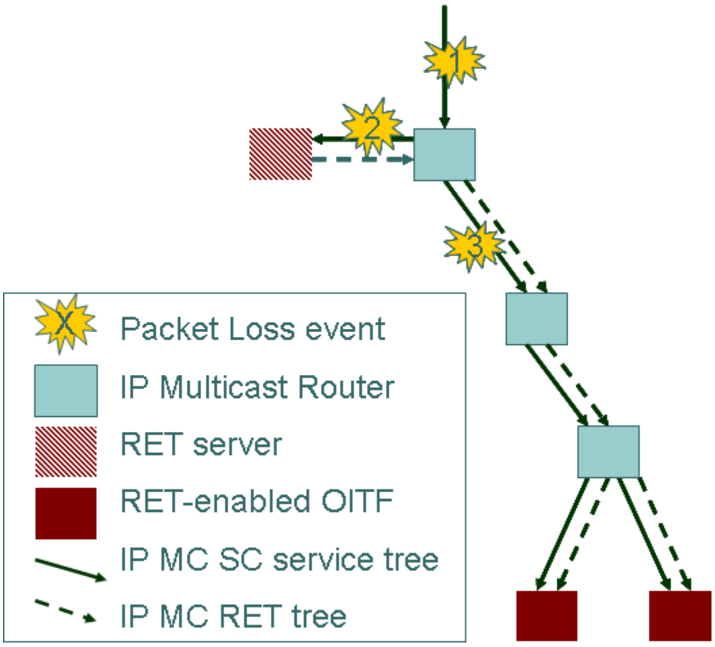


Figure 23: Example showing location of packet loss deep in the network relative to the RET server

M.1.3.1 Packet loss event upstream of the RET server

The packet loss events indicated as “1” and ‘2” in Figure 23 are examples of packet loss events upstream of the RET server.

The RET server is able to detect such packet loss event as there are missing packets in the IP multicast streams received by the RET server. Upon such packet loss detection, the RET server transmits a RTCP FB NACK on the IP multicast RET towards the RET-enabled OITFs.

Those OITFs that were instructed with SD&S to wait a random or fixed period before sending a (unicast) RTCP FB NACK upon packet loss detection (i.e. non-immediate reporters), SHALL abstain from sending a unicast RTCP FB NACK when receiving the RTCP FB NACK in the IP multicast RET within this waiting period. This mechanism allows to prevention or mitigation of NACK storms.

Note: DVB SD&S contains attributes for multicast RET by which it can signal to OITFs the duration of the waiting period, if any, and this waiting period can be different per OITF.

When the RET server is capable of recovering the missing packet(s), these are sent as retransmission packet(s) over the RET Multicast. How the RET server performs packet loss recovery for upstream packer loss is not addressed in this specification.

Note that the RET server in general can not distinguish between packet loss event 1 and packet loss event 2. therefore sending IP multicast RET packets is only effective when the packet loss event took place on a branch of the content IP multicast tree that also feeds the branches to which the OITFs serviced by this RET server are connected (packet loss event 1 in the figure). This means that in the case of packet loss event 2, the multicast RET packets are transmitted unnecessarily to the OITFs. The OITF SHALL discard RET packets when the corresponding original RTP packets have been received in the IP multicast content stream.

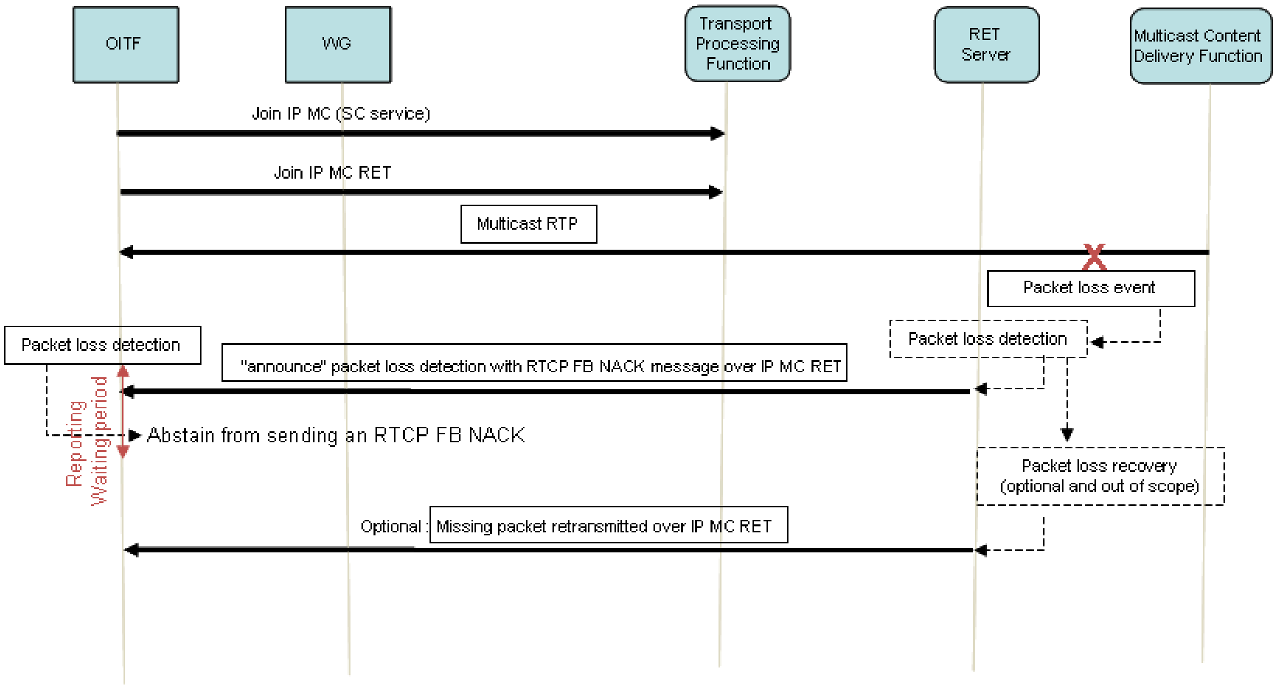

OITF
WG
Transport Processing Function
RET Server
Multicast Content Delivery Function
Join IP MC (SC service)
Join IP MC RET
Multicast RTP
Packet loss event
Packet loss detection
Packet loss detection
"announce" packet loss detection with RTCP FB NACK message over IP MC RET
Reporting Waiting period
Abstain from sending an RTCP FB NACK
Packet loss recovery (optional and out of scope)
Optional : Missing packet retransmitted over IP MC RET

Figure 24: Multicast RET protocol interaction for packet loss upstream of RET server

M.1.3.2 Packet loss event downstream of the RET server

The packet loss event “3” in Figure 23 is illustrative for a packet loss event downstream of the RET server and impacting several OITFs.

When there is a packet loss event on the IP multicast content service tree branch downstream of the RET server, impacting a large subset of OITFs, the RET server MAY indirectly detect this by receiving a large amount of RET requests from OITFs for the same missing packet(s). Those will be OITFs configured with SD&S as immediate reporters (no waiting period)

The RET server MAY then resend the packet(s) reported as missing RET packet(s) over the IP multicast RET.

A RET-enabled OITF that was impacted by the packet loss AND receives the IP multicast RET packet(s) during its reporting waiting period SHALL NOT send a unicast RTCP FB NACK.

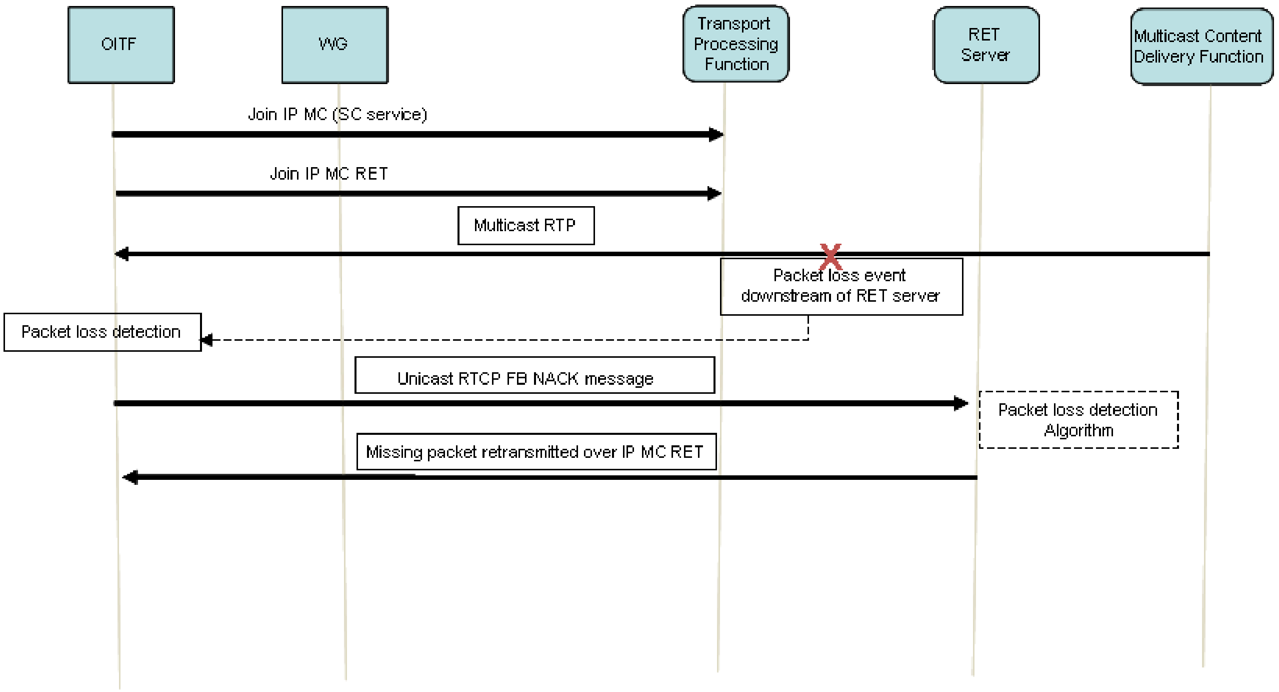

OITF
WG
Transport Processing Function
RET Server
Multicast Content Delivery Function
Join IP MC (SC service)
Join IP MC RET
Multicast RTP
Packet loss event downstream of RET server
Packet loss detection
Unicast RTCP FB NACK message
Packet loss detection Algorithm
Missing packet retransmitted over IP MC RET

Figure 25: Multicast RET protocol interaction for packet loss downstream of RET server, with the OITF acting as immediate reporter

The RET server MAY be provisioned to NOT service unicast retransmission requests from non-immediate reporting OITFs for packet losses that have been addressed by the RET server by means of sending an RTCP FB NACK or the retransmission packet over multicast RET. However, as packet loss events in the multicast RET MAY trigger such unicast retransmission requests, this is a service provider-dependent decision and capabilities of the RET server in terms of keeping track of each OITF's requesting behaviour and/or redundancy measures in the multicast RET will steer that decision.

M.2 Fast Channel Change (FCC)

The FCC solution is an OPTIONAL add-on service to the multicast content service (Scheduled Content Service), based on a server-based technique allowing minimization of the time between the request from a user for a new channel and the instant when the new channel starts displaying.

FCC is defined only for combination with multicast content service carried over RTP.

The FCC-enabled OITF interacts with a FCC server that receives and caches the IP multicast content streams for a limited time, and acts simultaneously as

- the unicast feedback target (FT) entity for the RTP IP multicast session transporting the content service, where the OITF sends its RTCP messages, including the FCC requests, in unicast to this FT entity.
- a burst source transmitting the FCC packets from its cache in a dedicated unicast RTP session.

When a multicast content service is both RET and FCC-enabled, the RET server and FCC server SHALL be one and the same.

When an end-user zaps to a channel that is FCC–enabled, and the OITF supports FCC, the following steps take place:

Step 0: The OITF leaves the “previous” IP multicast stream of the content service that MAY still be received when an end-user zaps (away) to the new FCC-enabled multicast content service.

Step 1: The FCC client of the OITF requests a fast channel change by sending a RTCP RAMS-R Feedback (FB) message to the FT entity of the FCC server. When the OITF performed the port mapping message exchange process, it also transmits the previously assigned cookie.

Step 2: The FCC/RET server responds with a RTCP RAMS-I FB message indicating that it accepts or does not accept the FCC request. This message also contains metadata that describes the data burst when the FCC server accepted the FCC request.

Step 3: The burst source entity of the FCC server starts sending RTP data (unicast stream) to the OITF from its cache and formatted as defined in [RFC4588]. The FCC server sends the unicast data to the client in general at a higher bitrate than the nominal bitrate of the incoming multicast stream. Video display at the client starts as soon as sufficient data is received in the buffer.

Step 4: As the FCC/RET server sends the last data from the buffered IP multicast content, the burst source entity signals the OITF by means of RTCP RAMS-I that it has to join the multicast stream (this message is OPTIONAL as the FCC server MAY have been forward looking and include this information in the RAMS-I message of step 2).

Step 5: The OITF SHALL send an IGMP/MLP join message to the TPF to receive the IP multicast stream (as defined in section 8.1.1, “Multicast content streaming service on UNIS-13”).

Step 6: The OITF receives the IP multicast stream.

Step 7: When OITF receives the first multicast packet, it sends an RTCP RAMS-T FB message to the burst source entity of the FCC server with the RTP sequence number of that packet such that the FCC server knows when to stop forwarding the incoming multicast data in the unicast FCC RTP session.

When the end-user selects a new channel, the following steps take place:

Step 8: The OITF issues an IGMP/MLP leave for the “old” channel (in line with the multicast content service specification) once it has executed step 5.

Step 9: The OITF sends an RTCP Bye message to the burst source entity of the FCC server to indicate it is leaving the unicast RTP session. The OITF also sends a (unicast) Bye message to the FT entity of the FCC serverto indicate it is leaving the “old” IP Multicast RTP session once it has executed step 5.

Then the FCC procedure for an FCC-enabled channel starts again with steps 1, 2, etc.

The FCC server will send the unicast RTP FCC packets to the OITF that issued the FCC request, but it needs to determine on which UDP port the OITF expects the unicast RTP FCC packets. There are two methods by which the OITF indicates to the FCC server its receive port for the unicast stream of RTP FCC packets:

- Cookie signaling

  This method is defined in [PORTMAP] and the procedure is as follows:

  a) The OITF sends a port mapping RTCP (request) message to the burst source entity of the FCC server. The source port of this message indicates to the FCC server the OITF receive port for FCC RTP packets

  b) The burst source entity of the FCC server responds with a port mapping response RTCP message containing a cookie, which SHALL be sent by the OITF in a dedicated RTCP message together with the RAMS-R RTCP message to the FT entity of the FCC server when requesting the FCC service.

  The FCC server processes the cookie received with the RTCP RAMS-R from the OITF, and this way, it knows the port for the subsequent FCC RTP packet flow to be transmitted to that OITF.

  This RTCP port mapping message exchange SHALL take place prior to sending the RAMS-R message, and SHALL be performed by the OITF each time the end-user zaps to a new channel that is FCC-enabled when cookie signaling MUST be used.

  If an OITF makes use of both the RET and FCC services for the multicast content service, there is a single port mapping message exchange process, as FCC and RET packets are transmitted in the same RTP session by the FCC/RET server. Hence the same cookie is transmitted along with the RTCP FCC request and RET request (NACK) RTCP messages, where the cookie remains valid until the OITF disconnects from the IP multicast stream.

- Without cookie signaling

  With this method the source port of the RAMS-R RTCP FB message indicates to the FCC server the OITF receive port for FCC RTP packets. The FCC server SHALL also assure that the source port of the FCC packets matches the FCC server receive port for the RTCP RAMS-R messages.

Note 1: A FCC-enabled OITF SHALL support RTP/RTCP muxing, i.e. the OITF SHALL be able to receive the RTP and RTCP packets in the unicast FCC session on the same receive port.

Note 2: The unicast FCC packet flow will traverse any NAT between OITF and FCC server because the NAT is primed by either the RTCP port mapping request message (cookie signaling approach) or by the RTCP RAMS-R message (without cookie signaling). See Annex G.3, “Port mapping and NAT traversal for FCC/RET for multicast content services”.

SD&S signals whether the port mapping messaging process (cookie signaling method) MUST be used or not by the OITF for the RET/FCC-enabled multicast content service. The cookie signaling method SHALL be supported by the OITF.

Figure 26 shows schematically the steps involved when an OITF makes use of the FCC service for multicact content service. Steps a and b are only present when the cookie signalling method is used.

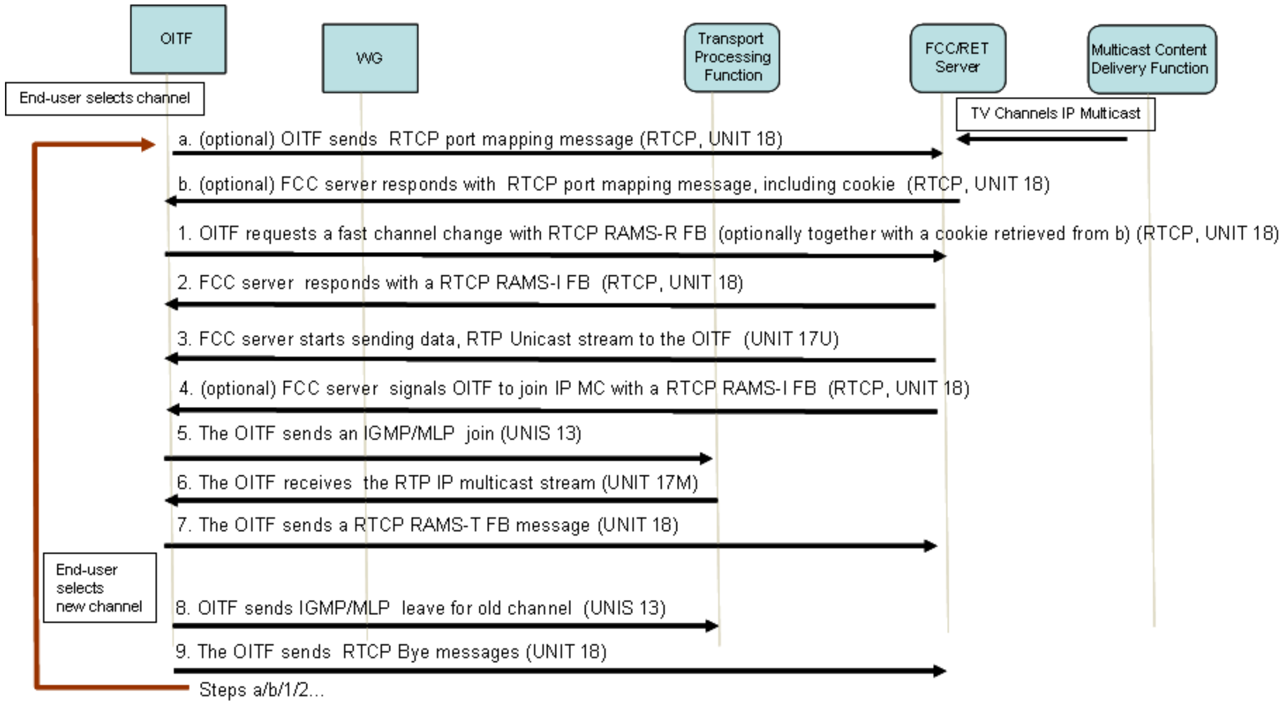

OITF
WG
Transport Processing Function
FCC/RET Server
Multicast Content Delivery Function
End-user selects channel
TV Channels IP Multicast
a. (optional) OITF sends RTCP port mapping message (RTCP, UNIT 18)
b. (optional) FCC server responds with RTCP port mapping message, including cookie (RTCP, UNIT 18)
1. OITF requests a fast channel change with RTCP RAMS-R FB (optionally together with a cookie retrieved from b) (RTCP, UNIT 18)
2. FCC server responds with a RTCP RAMS-I FB (RTCP, UNIT 18)
3. FCC server starts sending data, RTP Unicast stream to the OITF (UNIT 17U)
4. (optional) FCC server signals OITF to join IP MC with a RTCP RAMS-I FB (RTCP, UNIT 18)
5. The OITF sends an IGMP/MLP join (UNIS 13)
6. The OITF receives the RTP IP multicast stream (UNIT 17M)
7. The OITF sends a RTCP RAMS-T FB message (UNIT 18)
End-user selects new channel
8. OITF sends IGMP/MLP leave for old channel (UNIS 13)
9. The OITF sends RTCP Bye messages (UNIT 18)
Steps a/b/1/2...

Figure 26: Call flow for Fast Channel Change Service

Note: The RTCP FCC control messages are defined in Unicast-Based Rapid Acquisition of Multicast RTP Sessions [RAMS].

Annex N IG handling of IMPUs in association with GRUU (informative)

3GPP has mandated the usage of GRUU for certain features such as session transfer. Support for GRUU has impacts on the OITF and the IG.

The OITF impacts involve requesting a GRUU from the network.

Support in the IG is more complex due to the fact how GRUU works. The following is a brief description of rationale for the impacts in the IG.

The URI in the contact header used for registering a user is made up of user@host, where user is the username or IMPU and the host is the IP address of the IG.

According to [RFC5627], a GRUU returned from the network to an OITF requesting a GRUU, includes the sip instance feature tag for the OITF. That GRUU can be later used to address that specific user on the specific device (OITF). Since the same user registered from multiple OITF devices will have a distinct sip instance per OITF, the GRUU allocated by the network to each user on each OITF is different. As such, a request can be explicitly targeted to a specific user on a specific device.

However, using GRUU requires some special handling in the IG. The problem arises from how GRUU is processed by the network; a GRUU deployed to address a user in an incoming request is dereferenced by the network to the user contact registered for the specific user before the request is delivered to the user. Typically if there is no IG, each contact will have a distinct IP address, hence there is no ambiguity. However given that users in a residential LAN are behind the same IG, hence they the same IP address is used in all the contacts. As such if the same user is registered from multiple devices, the same contact will be used, even though the GRUU is different, but because the network dereferences the GRUU to the user contact, there is no way for the IG of selecting the right user. For an incoming request.

To remedy this, the IG MUST allocate for every user a distinctive virtual user name that can be placed in the user portion for the user@host portion when it registers the user with the IMS network. When an incoming request arrives to the IG, the IG will be able to identify the right target device and consequently the correct OITF. The IG maintains a binding between the real username submitted by the OITF and the allocated virtual name used by the IG for registration purpose for the entire time the user is registered.

Annex O FDT Schema Extensions

This section describes the FDT schema extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:fdt:2010"
    xmlns:fl="http://www.example.com/flute"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.example.com/flute"
   schemaLocation="imports/Flute_FDT.xsd"/>
  <xs:element name="FDT-Instance">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="fl:CommonFDTAttributes">
          <xs:sequence>
            <xs:element name="File" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="fl:FileCommonFDTAttributes">
                   <xs:attribute name="Tag" type="xs:string" use="optional"/>
                   <xs:attribute name="Priority" type="xs:int" use="optional"/>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Annex P IG Service Awareness

In its role as a B2BUA facing the OITF on one hand and the IMS network on the other hand, and for the IG to insert the appropriate SIP headers reflecting the service desired by the OITF, the IG must be able to detect from the incoming request the service desired by the OITF.

Following the detection of the requested service, the IG SHALL than insert the additional SIP headers with the appropriate values. For some of the requested services, such as multimedia telephony, and some IPTV related services, the additional information is defined in the appropriate specification defining those services.

For other services, the IG can be configured for any additional information needed if the services require that. Some services may not require any additional SIP headers to be inserted by the IG.

P.1 IG Algorithm for service awareness

The IG SHALL support the following algorithm to allow it to determine the service desired by an OITF, and subsequently the additional SIP headers. This algorithm applies only to OITF-initiated sessions:

- If the Request URI in the incoming INVITIE is the well known PSI for a content service, then the service desired by the OITF is a content service. Furthermore, if the content session is associated with a session transfer than the IG executes the procedure defined in section 6.2.2.9.1.2 “IG handling of session transfers when the transferor and transferee are behind the same IG”. The algorithm terminates
- If the Request URI in the incoming INVITE includes the PSI of a service configured in the IG, then the desired service is identified and the algorithm terminates
- If the Accept header in the incoming SIP INVITE is set to message/cpim, then the service desired by the OITF is IM chat, and the algorithm terminates. Note in this case the IG SHALL also ensure that the SDP include the key words “msrp” associated with the session chatting using MSRP
- The IG examines the SDP according to the following:
    - If the SDP includes one audio m-line only, then the service desired by the OITF is multimedia telephony (MMTEL) and the algorithm terminates
    - If the SDP includes one audio m-line and one or more video m-lines, then the service desired by the OITF is multimedia telephony (MMTEL) and the algorithm terminates
    - If the SDP includes attributes configured in the IG, and that can be matched against the configured service and the algorithm terminates.
    - The IG cannot determine the service, it shall proxy the information as received from the OITF without any additional SIP headers..

Annex Q Definition of Content-Reporting Info Package

This annex defines an Info Package [INFO-PKG] for sending information on content streamed by an end-user on a device using SIP INFO requests

Content-Reporting Info Package

Q.1 Overall General

This subclause contains the information required for the IANA registration of an Info Package.

Q.2 Overall Description

When an end-user is streaming content from a device such as an OITF, there is a desire to report the content streamed by the end-user to a central server, such an IPTV Control Functional Entity. This permits services, such as parental control, to be offered to users with the proper authority, and which allows intervention by those users to stop the streamed content, if desired, or take other appropriate actions. Reporting the streamed content does not require user intervention. It is performed autonomously by applications executing on the reporting device if configured to do so. The streamed content can be either a scheduled content or Content on demand, but can be extended to other types of content as well.

The Content-Reporting Info Package is used to transport the necessary information regarding the content streamed by a user. The Content-Reporting Info Package is used to transfer a single content streamed by a user at any time. As such, only one content is transported in a single SIP INFO request.

The Content-Reporting Info Package is defined for any multimedia application that includes content streaming. Any application, where sending information regarding content streamed by a user using the SIP INFO method is required, can use the Content-Reporting Info Package.

Q.3 Applicability

The Info Package mechanism for transporting information about streamed content has been chosen since this is a service that some networks may offer, and as such it is optional. Also networks that offer the service may want to selectively activate the service for some subscribers and not for others. Finally, due to the nature of some multimedia applications, the network needs to be able to selectively stop and /or resume the reporting of the streamed content from a device depending on policies in the network. This is due to the fact that for some multimedia applications, a large number of messages reporting streamed content may be sent simultaneously to the server leading to load problems.

Finally, the mechanism also allows information about streamed content to be sent inside an existing dialog, using the same signalling path as other SIP messages within the dialog, rather than having to establish a separate dialog.

Q.4 Info Package Name

The name of the Info Package is: Content-Reporting

Q.5 Info Package Parameters

No parameters are defined for the Content-Reporting Info Package.

Q.6 SIP Option Tags

No SIP option tags are defined for the Content-Reporting Info Package.

Q.7 INFO Message Body Parts

Q.7.1 General

Information on streamed content on a device is sent as part of the message body of the SIP INFO request. This subclause defines the information and syntax associated with the message body part used for transporting the information.

Q.7.2 SIP Content-Type header field value

The value of the SIP Content-Type header field associated with the Content-Reporting Info Package message body is: application/3gpp-ims-pss-mbms-command+xml”

Q.7.3 SIP Content-Disposition header field value

The value of the SIP Content-Disposition header field associated with the Content-Reporting Info Package message body is: Info-Package.

Q.7.4 Message body syntax

The syntax of the Content-Reporting Info Package message body is based on the rules defined in section 5.3.1.1.6.1

Q.8 Info Package Usage Restrictions

No usage restrictions are defined for the Content-Reporting Info Package.

Q.9 Rate of INFO Requests

No maximum rate or minimum rate is defined for sending INFO requests associated with the Content-Reporting Info Package.

When Content-Reporting requests are generated by devices, the package does not provide a feedback mechanism to indicate to the sender that the rate of Content-Reporting generation is too slow or fast. However applications in the network can order a device to stop reporting completely or resume reporting at any time if they so desire.

Q.10 Info Package Security Considerations

No additional security mechanism is defined for the Content-Reporting Info Package.

The security of the Content-Reporting Info Package is based on the generic security mechanism provided for the underlying SIP signalling.

Q.11 Implementation Details and Examples

None

Annex R Definition of Digital-Media-Purchase Info Package

This annex defines an Info Package [INFO-PKG] for sending information on content streamed by an end-user on a device using SIP INFO requests

Digital-Media-Purchase Info Package

R.1 Overall General

This subclause contains the information required for the IANA registration of an Info Package.

R.2 Overall Description

Digital purchases are triggered by end users, while watching streamed content. These purchases are associated with advertisements made available to end-users while watching streamed content.

The Digital-Media-Purchase Info Package is used to transport the necessary information regarding the digital content purchased by the end-user. The Digital-Media-Purchase Info Package is used to transfer purchase information regarding a single digital content. As such, purchase information related to only one digital content is transported in a single SIP INFO request.

The Digital-Media-Purchase Info Package is defined for any multimedia application that includes content streaming with advertisement for purchasing digital content. Any application, where sending information regarding digital content purchased bye an end-user using the SIP INFO method is required, can use the Digital-Media-Purchase Info Package.

R.3 Applicability

The Info Package mechanism for transporting information about purchased digital content by an end user has been chosen since this is a service that some networks may offer, and as such it is optional. Also networks that offer the service may want to selectively activate the service for some subscribers and not for others.

Finally, the mechanism also allows information about purchased digital content to be sent inside an existing dialog, using the same signalling path as other SIP messages within the dialog, rather than having to establish a separate dialog.

R.4 Info Package Name

The name of the Info Package is: Digital-Media-Purchase

R.5 Info Package Parameters

No parameters are defined for the Digital-Media-Purchase Info Package.

R.6 SIP Option Tags

No SIP option tags are defined for the Digital-Media-Purchase Info Package.

R.7 INFO Message Body Parts

R.7.1 General

Information on charges related to digital content purchased by an end-user is sent as part of the message body of the SIP INFO request. This subclause defines the information and syntax associated with the message body part used for transporting the information.

R.7.2 SIP Content-Type header field value

The value of the SIP Content-Type header field associated with the Digital-Media-Purchase Info Package message body is: application/vnd.oipf.purchase+xml”

R.7.3 SIP Content-Disposition header field value

The value of the SIP Content-Disposition header field associated with the Digital-Media-Purchase Info Package message body is: Info-Package.

R.7.4 Message body syntax

The syntax of the Digital-Media-Purchase Info Package message body is based on the rules defined in section 5.3.5.8, “XML Schema for Purchase Request of Digital Media”

R.8 Info Package Usage Restrictions

No usage restrictions are defined for the Digital-Media-Purchase Info Package.

R.9 Rate of INFO Requests

No maximum rate or minimum rate is defined for sending INFO requests associated with the Digital-Media-Purchase Info Package.

When Digital-Purchase requests are generated by end-user devices, the package does not provide a feedback mechanism to indicate to the sender that the rate of Digital-Purchase generation is too slow or fast. However applications in the network can stop and/or resume purchases related to digital content at any time.

R.10 Info Package Security Considerations

No additional security mechanism is defined for the Digital-Media-Purchase Info Package.

The security of the Digital-Media-Purchase Info Package is based on the generic security mechanism provided for the underlying SIP signalling.

R.11 Implementation Details and Examples

None

Annex S Definition of Parental-Control-Watched-Content Info Package

This annex defines an Info Package [INFO-PKG] for sending information on parental control watched content using SIP INFO requests

Digital-Media-Purchase Info Package

S.1 Overall General

This subclause contains the information required for the IANA registration of an Info Package.

S.2 Overall Description

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.

The Parental-Control-Watched-Content Info Package is used to transport the necessary information regarding the parental control watched content in a single SIP INFO request.

The Parental-Control-Watched-Content Info Package is defined for any multimedia application that includes parental control watched content. Any application, where sending parental control watched content using the SIP INFO method is required, can use the Parental-Control-Watched-Content Info Package.

S.3 Applicability

The Info Package mechanism for transporting information about parental control watched content has been chosen since this is a service that some networks may offer, and as such it is optional. Also networks that offer the service may want to selectively activate the service for some subscribers and not for others.

Finally, the mechanism also allows information about parental control watched content to be sent inside an existing dialog, using the same signalling path as other SIP messages within the dialog, rather than having to establish a separate dialog.

S.4 Info Package Name

The name of the Info Package is: Parental-Control-Watched-Content

S.5 Info Package Parameters

No parameters are defined for the Parental-Control-Watched-Content Info Package.

S.6 SIP Option Tags

No SIP option tags are defined for the Parental-Control-Watched-Content Info Package.

S.7 INFO Message Body Parts

S.7.1 General

Information on parental control watched content is sent as part of the message body of the SIP INFO request. This subclause defines the information and syntax associated with the message body part used for transporting the information.

S.7.2 SIP Content-Type header field value

The value of the SIP Content-Type header field associated with the Parental-Control-Watched-Content Info Package message body is: application/vnd.oipf.parental-control+xml.

S.7.3 SIP Content-Disposition header field value

The value of the SIP Content-Disposition header field associated with the Parental-Control-Watched-Content Info Package message body is: Info-Package.

S.7.4 Message body syntax

The syntax of the Parental-Control-Watched-Content Info Package message body is based on the rules defined in section 5.3.7.1.4, " XML Schema for Parental Control Watched Content".

S.8 Info Package Usage Restrictions

No usage restrictions are defined for the Parental-Control-Watched-Content Info Package.

S.9 Rate of INFO Requests

No maximum rate or minimum rate is defined for sending INFO requests associated with the Parental-Control-Watched-Content Info Package.

When Parental-Control-Watched-Content requests are generated by end-user devices, the package does not provide a feedback mechanism to indicate to the sender that the rate of Parental-Control-Watched-Content generation is too slow or fast. However applications in the network can stop and/or resume purchases related to digital content at any time.

S.10 Info Package Security Considerations

No additional security mechanism is defined for the Parental-Control-Watched-Content Info Package.

The security of the Parental-Control-Watched-Content Info Package is based on the generic security mechanism provided for the underlying SIP signalling.

S.11 Implementation Details and Examples

None

Annex T Common Types

T.1 Schema

The following XML schema defines common types used in this specification

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

  <xs:complexType name="ProgramIdType">
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attribute name="ProgramType" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="UserIdType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

T.2 Definitions

T.2.1 ProgramIdType

This datatype is used to define elements which convey the identifier of a program. The ProgramType attribute SHALL be either “SC” for a program delivered as part of the Scheduled Content service, or “COD” for a program delivered as part of the Content On Demand service.

T.2.2 UserIdType

This datatype is used to define elements which convey the identity of a user. Additional syntax, formatting or restriction of the information carried may be defined in this specification

Annex U Schema Extension for FLUTE FDT

U.1 Namespace

The namespace for Open IPTV Forum is “urn:oipf:protocol:fluteFDT:2009”.

```
<schema xmlns:tns="urn:oipf:protocol:fluteFDT:2009"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:fl="http://www.example.com/flute"
targetNamespace="urn:oipf:protocol:fluteFDT:2009" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!-- schema filename is  protocol-fluteFDT.xsd -->
```

U.2 Import Namespace and schema

```
<import namespace="http://www.example.com/flute"
  schemaLocation="imports/Flute_FDT.xsd"/>
```

U.3 Extension of FDT Attributes

When FLUTE is used for delivery of objects via multicast the FDT-Instance XML structure is extended using the extension mechanism defined in [FLUTE].

The following attributes are added to the FDT-Instance element, after the standard attributes.

```
<complexType name="OIPFCommonFDTAttributes">
  <complexContent>
    <extension base="fl:CommonFDTAttributes">
      <attribute name="Tags" type="string" use="required">
        <annotation>
          <documentation>
            This string value contains multiple tags, delimited with a
            semicolon (";"), that describe the File
            e.g. "Tag1;Tag2;Tag Three;"
          </documentation>
        </annotation>
      </attribute>
      <attribute name="Priority" type="positiveInteger" use="optional"
          default="10"/>
    </extension>
  </complexContent>
</complexType>
```