**IIF-WT-063R44**

DRAFT Working Text on

# IPTV Content on Demand Service

Secretariat

**Alliance for Telecommunications Industry Solutions**

Approved Month DD, YYYY

**Abstract**

Abstract text here.

## Foreword

The Alliance for Telecommunication Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The IPTV Interoperability Forum (IIF) enables the interoperability, interconnection, and implementation of IPTV systems/services by developing ATIS standards and facilitating related technical activities. This forum will place an emphasis on North American and ATIS Member Company needs in coordination with other regional and international standards development organizations.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, IIF Secretariat, 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time it approved this document, IIF, which is responsible for the development of this standard, had the following members:

[COMMITTEE LIST]

The ARCH Committee was responsible for the development of this document.

## Notice of Disclaimer & Limitation of Liability (delete on PUB; on back of Cover)

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith.

## Revision History (delete on PUB)

| Date | Version | Description | Author |
|---|---|---|---|
| 2/19/09 | R0 | Incorporates the following contributions:<br>IIF-ARCH-2009-026R2<br>IIF-ARCH-2009-050R2<br>IIF-ARCH-2009-081<br>IIF-ARCH-2009-083<br>IIF-ARCH-2009-084 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 2/20/09 | R1 | IIF-ARCH-2009-091 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 3/17/09 | R2 | IIF-ARCH-2009-148R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 4/14/09 | R3 | IIF-ARCH-2009-211 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 4/23/09 | R4 | IIF-ARCH-2009-244R2<br>IIF-ARCH-2009-245R2<br>IIF-ARCH-2009-256 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 5/19/09 | R5 | IIF-ARCH-2009-311 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 6/2/09 | R6 | IIF-ARCH-2009-244R4<br>IIF-ARCH-2009-275R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 7/28/09 | R7 | IIF-ARCH-2009-432R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 8/13/09 | R8 | IIF-ARCH-2009-441<br>IIF-ARCH-2009-442<br>IIF-ARCH-2009-445R1<br>IIF-ARCH-2009-461R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 8/25/09 | R9 | IIF-ARCH-2009-484R2<br>IIF-ARCH-2009-488<br>IIF-ARCH-2009-494R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 9/8/09 | R10 | IIF-ARCH-2009-337R2<br>IIF-ARCH-2009-503R1<br>IIF-ARCH-2009-518R2<br>IIF-ARCH-2009-521R1<br>IIF-ARCH-2009-523R1<br>IIF-ARCH-2009-524R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 9/23/09 | R11 | IIF-ARCH-2009-542R1<br>IIF-ARCH-2009-544R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 10/26/09 | R12 | IIF-ARCH-2009-543 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |

| | | | |
|---|---|---|---|
| 11/20/09 | R13 | IIF-ARCH-2009-626R1<br>IIF-ARCH-2009-629R1<br>IIF-ARCH-2009-630<br>IIF-ARCH-2009-646<br>IIF-ARCH-2009-669<br>IIF-ARCH-2009-675R1<br>IIF-ARCH-2009-678 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 1/6/10 | R14 | IIF-ARCH-2010-004 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 2/11/10 | R15 | Meeting Edits | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 2/11/10 | R16 | IIF-ARCH-2009-628R3<br>IIF-ARCH-2010-027R1<br>IIF-ARCH-2010-061R2<br>IIF-ARCH-2010-065R1<br>IIF-ARCH-2010-066R1<br>IIF-ARCH-2010-070R2<br>IIF-ARCH-2010-083<br>IIF-ARCH-2010-085 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 2/24/10 | R17 | IIF-ARCH-2009-083<br>IIF-ARCH-2010-099R1<br>IIF-ARCH-2010-118 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 3/24/10 | R18 | IIF-ARCH-2010-113R1<br>IIF-ARCH-2010-142<br>IIF-ARCH-2010-157<br>IIF-ARCH-2010-158<br>IIF-ARCH-2010-159R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 3/30/10 | R19 | IIF-ARCH-2010-166R1<br>IIF-ARCH-2010-177R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 3/31/10 | R20 | Live edits during Ashburn interim meeting | ARCH Committee |
| 3/31/10 | R21 | IIF-ARCH-2010-180R2<br>IIF-ARCH-2010-188<br>IIF-ARCH-2010-190<br>IIF-ARCH-2010-191R1<br>IIF-ARCH-2010-192R1<br>IIF-ARCH-2010-193<br>IIF-ARCH-2010-194 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 4/7/10 | R22 | IIF-ARCH-2010-200<br>IIF-ARCH-2010-207<br>IIF-ARCH-2010-208 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 4/23/10 | R23 | IIF-ARCH-2010-178R4<br>IIF-ARCH-2010-209R4<br>IIF-ARCH-2010-249R1<br>IIF-ARCH-2010-252R2<br>IIF-ARCH-2010-255R1<br>IIF-ARCH-2010-256 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |

| | | | |
|---|---|---|---|
| | | IIF-ARCH-2010-257R1<br>IIF-ARCH-2010-260R1<br>IIF-ARCH-2010-261R1 | |
| 5/5/10 | R24 | IIF-ARCH-2010-285 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 5/14/10 | R25 | IIF-ARCH-2010-273R1<br>IIF-ARCH-2010-278R2<br>IIF-ARCH-2010-279R1<br>IIF-ARCH-2010-291R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 5/19/10 | R26 | IIF-ARCH-2010-284R3<br>IIF-ARCH-2010-313R2 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 6/2/10 | R27 | IIF-ARCH-2010-339<br>IIF-ARCH-2010-350R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 6/16/10 | R28 | IIF-ARCH-2010-364R2<br>IIF-ARCH-2010-380<br>IIF-ARCH-2010-384<br>IIF-ARCH-2010-385<br>IIF-ARCH-2010-386<br>IIF-ARCH-2010-391 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 6/30/10 | R29 | IIF-ARCH-2010-395<br>IIF-ARCH-2010-403R1<br>IIF-ARCH-2010-405R1<br>IIF-ARCH-2010-410R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 7/14/10 | R30 | IIF-ARCH-2010-359R5<br>IIF-ARCH-2010-360R2<br>IIF-ARCH-2010-394R1<br>IIF-ARCH-2010-409R1<br>IIF-ARCH-2010-432<br>IIF-ARCH-2010-445R1<br>IIF-ARCH-2010-446R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 7/28/10 | R31 | IIF-ARCH-2010-363R3<br>IIF-ARCH-2010-430<br>IIF-ARCH-2010-447R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 8/11/10 | R32 | IIF-ARCH-2010-415R3<br>IIF-ARCH-2010-485<br>IIF-ARCH-2010-486<br>IIF-ARCH-2010-503<br>IIF-ARCH-2010-510 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 8/17/10 | R33 | Meeting edits<br>IIF-ARCH-2010-513R1<br>IIF-ARCH-2010-529R1<br>IIF-ARCH-2010-530<br>IIF-ARCH-2010-535<br>IIF-ARCH-2010-539R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 8/19/10 | R34 | Meeting edits | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 8/19/10 | R35 | IIF-ARCH-2010-511R1<br>IIF-ARCH-2010-518 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |

| | | | |
|---|---|---|---|
| | | IIF-ARCH-2010-528R1<br>IIF-ARCH-2010-548R1<br>IIF-ARCH-2010-549R1<br>IIF-ARCH-2010-556R1<br>IIF-ARCH-2010-559<br>IIF-ARCH-2010-560 | |
| 9/1/10 | R36 | IIF-ARCH-2010-577<br>IIF-ARCH-2010-580R1<br>IIF-ARCH-2010-581<br>IIF-ARCH-2010-584R1<br>IIF-ARCH-2010-585R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 9/15/10 | R37 | IIF-ARCH-2010-579R1<br>IIF-ARCH-2010-614R1 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 9/29/10 | R38 | IIF-ARCH-2010-632<br>IIF-ARCH-2010-637<br>IIF-ARCH-2010-640R2<br>IIF-ARCH-2010-644R1<br>IIF-ARCH-2010-645 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 10/13/10 | R39 | IIF-ARCH-2010-658<br>IIF-ARCH-2010-659<br>IIF-ARCH-2010-665<br>IIF-ARCH-2010-666<br>IIF-ARCH-2010-667<br>IIF-ARCH-2010-668R1<br>IIF-ARCH-2010-669<br>IIF-ARCH-2010-677 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 10/27/10 | R40 | IIF-ARCH-2010-694<br>IIF-ARCH-2010-695R1<br>IIF-ARCH-2010-696R1<br>IIF-ARCH-2010-702<br>IIF-ARCH-2010-703<br>IIF-ARCH-2010-705<br>IIF-ARCH-2010-707 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |
| 11/8/10 | R41 | Meeting edits | ARCH Committee |
| 11/9/10 | R42 | IIF-ARCH-2010-747<br>IIF-ARCH-2010-753<br>Meeting edits | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon<br>ARCH Committee |
| 11/10/10 | R43 | IIF-ARCH-2010-749R1<br>IIF-ARCH-2010-751R1<br>Meeting edits | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon<br>ARCH Committee |
| 11/11/10 | R44 | IIF-ARCH-2010-745R2 | Mike Hluchyj, Verivue<br>Reza Shafiee, Verizon |

TABLE OF CONTENTS

## Table of Figures

## Table of Tables

Working Text on –

# IPTV Content on Demand Service

## 1 PURPOSE & SCOPE

### 1.1 Purpose

The purpose of this document is to describe the use of the relevant functions identified in ATIS-0800007 [10], *IPTV High Level Architecture*, for delivery of an IPTV Content on Demand (CoD) Service and developed in other ATIS IIF specifications. An instance of an IPTV CoD Service may be configured to provide a consumer experience similar to that of traditional Video on Demand (VoD) television services, but access to the potentially greater functionality also available through capabilities of the IPTV infrastructure.

This IPTV CoD Service Specification has been developed as a partial response to the requirements identified in ATIS-0800002 [8], *IPTV Architecture Requirements,* which grouped them as Pay Per View (PPV), VoD or download-based video content distribution requirements. This specification is a partial response as the standardization work has been reduced in scope by the ATIS-0800003 [9], *IPTV Architecture Roadmap*, to reflect industry consensus views of evolving priorities within these requirement areas.

#### 1.1.1 PPV Requirements

| | |
|---|---|
| IIF.ARCH.SERVICE.05 | The IPTV Architecture shall provide mechanisms to support PPV services. |

#### 1.1.2 VoD Requirements

| | |
|---|---|
| IIF.ARCH.SERVICE.06 | The IPTV Architecture shall provide mechanisms to support On Demand services. |

| | |
|---|---|
| IIF.ARCH.SERVICE.07 | The IPTV Architecture shall provide the capability for management of capacity on the services and network elements. |

| | |
|---|---|
| IIF.ARCH.SERVICE.08 | The IPTV Architecture shall support multiple service models for VoD such as Subscription VoD (SVoD) and Free VoD (FVoD). |

| | |
|---|---|
| IIF.ARCH.SERVICE.09 | The IPTV Architecture shall enable the service provider to configure the packages and package types available to the consumer. |

| | |
|---|---|
| IIF.ARCH.SERVICE.10 | The IPTV Architecture shall allow the delivery of multiple VoD content profiles -- e.g., HD, SD, multi-channel audio. |

| | |
|---|---|
| IIF.ARCH.SERVICE.11 | The IPTV Architecture shall make it possible for the user to restrict purchases and access to services and content through the use of appropriate controls -- e.g., PIN number, login. |

| | |
|---|---|
| IIF.ARCH.SERVICE.12 | The IPTV Architecture shall allow for the insertion of content into VoD content. |

| | |
|---|---|
| IIF.ARCH.SERVICE.13 | The IPTV Architecture shall support 3rd party content providers. |

| | |
|---|---|
| IIF.ARCH.SERVICE.14 | The IPTV Architecture may provide mechanisms to capture and utilize user profiles and preferences to target/restrict content items. |

| | |
|---|---|
| IIF.ARCH.SERVICE.15 | The IPTV Architecture shall support the acquisition of appropriate VoD accounting data, to fulfill licensing agreements. |

### 1.1.3 Download-based Video Content Distribution Requirements

| | |
|---|---|
| IIF.ARCH.SERVICE.21 | The IPTV Architecture shall provide mechanisms to support pre-positioned content services -- e.g., Push VoD. |

## *1.2 Scope*

This specification defines the basic IPTV CoD Service operation after the initialization, configuration, service provider discovery, and services discovery documented in ATIS-0800017 [13], *Network Attachment and Initialization of Devices and Client Discovery of IPTV Services* and ATIS-0800009 [11], *Remote Management of Devices in the Consumer Domain for IPTV Services*. Note that those specifications may enable an IPTV Terminal Function (ITF) to access additional IPTV services beyond the basic IPTV CoD Service described in this specification. This specification assumes that the ITF is authorized to access one or more service packages. Mechanisms to restrict or modify the set of available authorized service packages are out of the scope of this specification. Refer to EPG Metadata Specification (ATIS-0800020.v002 [15]).

Further distribution beyond the ITF, user interface controls for presentation on a television set, etc., are out of the scope of this specification. The IPTV CoD Service delivers content that may be either secured or unsecured. Where content security is required to ensure confidentiality of the content and constrain its further usage or distribution, appropriate Digital Rights Management mechanisms are to be used. These mechanisms are out of the scope of this specification.

For this specification, the content to be distributed on demand is restricted to be delivered over a *unicast* transport to the ITF. Multicast or broadcast IP service infrastructures[1] are considered out of scope for CoD service delivery to the ITF in this specification.

This specification is agnostic of infrastructures below the IP layer. This specification can be used for low bandwidth and/or wireless infrastructures, but it is not optimized or explicitly addressed for those purposes.

This specification describes a basic acquisition mechanism for IPTV CoD with options for various network resource allocation mechanisms.

In this release, the Content Distribution and Delivery Functions are administered by the IPTV service provider.

## 2 NORMATIVE REFERENCES

The following standards contain provisions which, through reference in this text, constitute provisions of this ATIS Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this ATIS Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

[1] 3GPP TS 23.228, *IP Multimedia Subsystem (IMS)*

[2] 3GPP TS 24.229, *IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3*

[3] 3GPP TS 29.228: *IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents*.

[4] 3GPP TS 29.229: *Cx and Dx Interfaces based on the Diameter protocol, Protocol details*.

[5] 3GPP TS 29.328, *IP Multimedia Subsystem (IMS) Sh interface; Signalling flows and message contents*

[6] 3GPP TS 29.329, *Sh interface based on the Diameter protocol; Protocol details*

[7] ATIS-0800001.v002, *IPTV DRM Interoperability Requirements, May 2007*

[8] ATIS-0800002, *IPTV Architecture Requirements, May 2006*.

[9] ATIS-0800003, *IPTV Architecture Roadmap, August 2006*

[10] ATIS-0800007, *IPTV High Level Architecture, April 2007.*

[11] ATIS-0800009.v002, *Remote Management of Devices in the Consumer Domain for IPTV Services, September 2009*

[12] ATIS-0800013, *Media Formats and Protocols for IPTV Services, January 2009.*

[13] ATIS-0800017.v002, *Network Attachment and Initialization of Devices and Client Discovery of IPTV Services, April 2009.*

[14] ATIS-0800018, *IPTV Linear TV Service, January 2009*

[1] Some VoD or NVoD services have been proposed over broadcast or multicast transport infrastructure.

[15] ATIS-0800020.v002, *IPTV Electronic Program Guide Metadata Specification, xxxx 2010?*
[16] ATIS-0800037, *IPTV Device and Authentication Identity Interoperability Specification* (WT67)
[17] ATIS-0800043, *Content on Demand Metadata Schema and Metadata Transactions* (WT71)
[18] ATIS-0800044, *IPTV Media Bookmark Specification* (WT80)
[19] ISO/IEC 13818-6 *Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC*
[20] ITU-T Y.1541, *Network Performance Objectives for IP-based Services.*
[21] ITU-T Y.1910, *IPTV Functional Architecture*
[22] ITU-T Y.2014, *Network Attachment Control Functions in Next Generation Networks*
[23] ITU-T Y.2111, *Resource and Admission Control Functions in Next Generation Network*s
[24] RFC 2326, *Real Time Streaming Protocol (RTSP*)
[25] RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*
[26] RFC 2965, *HTTP State Management Mechanism*
[27] RFC 3305, *Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations*.
[28] RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*
[29] RFC 4145, *TCP-Based Media Transport in the Session Description Protocol (SDP)*
[30] RFC 5988, *Web Linking*

# 3 Definitions, Acronyms, & Abbreviations

## 3.1 Definitions

### 3.1.1 Asset

A managed atomic resource with a globally unique opaque identity consisting of an OriginContentId and an associated state which is used in the implementation of a service. The Asset state consists of metadata, and may also contain Content. The Metadata may include references to other Assets.

The specification of allowable tags and values for Metadata items in Assets depends on which reference points the Assets are communicated: A3, A7, C1, C5, or E1.

### 3.1.2 Asset Source

An entity from which Assets may come that may be external to the Service Provider. Assets may also be created by functions within the Service Provider's system.

### 3.1.3 Catalog

EPG metadata for the CoD assets available for selection by the ITF.

### 3.1.4 Content

Data which is usually the media that is delivered to the ITF over the Ud interface. It typically consists of video or audio, but it is not limited to those types. See also *Media Resource*.

### 3.1.5 Continuous Media (adapted from RFC 2326):

A sub-category of Content, where there is a timing relationship between source and sink. The term "Continuous Media" does not apply to IPTV Linear TV service to the end-user.

### 3.1.6 Bookmark

A reference to content that may be viewed on a device implementing the ITF and which may include temporal information identifying a point or segment within the content. Bookmarks are classified in two distinct and separate manners: 1) by their origin and 2) by the temporal metadata media data included as specified in ATIS-0800044, *IPTV Media Bookmark Specification* [18].

### 3.1.7 Content Fragment

A segment of content identified as a separate resource with a unique identifier.  Content may be divided into fragments for the sake of manipulation and transfer to or retrieval by the ITF client. Content fragment identifiers associated with a single Content item may be listed in or derived from a media presentation description (e.g., playlist, manifest file).

### 3.1.8 Default User

A user who is primarily associated with a subscription by the terms of a subscription account. The default user may have administrative authority over all other users of subscription services. Attributes unique to a default user as identified by the UserId would be included in the Service User Profile.

### 3.1.9 Manifest File

A file that contains structural metadata used in the HTTP fragmented delivery of a media resource.

### 3.1.10 Media Resource

An instance of a resource as defined in RFC 2616, which is typically composed of multiple associated resources (e.g., normal and trick play continuous media, index data).

### 3.1.11 Media Resource Metadata

Media Resource Metadata defines the properties of the Media Resource entity body.  Media Resource Metadata *may* contain references to other Media Resources.

### 3.1.12 Media Resource Set

The collection of all of the resources referenced by Media Resource Metadata.

### 3.1.13 Normal Play Time (NPT)

The continuous timeline “clock” the viewer associates with a program, relative to the beginning of the program. See ISO/IEC 13818-6 [19], sections 5.5.1.3.2 & 8.1 for more information.

### 3.1.14 **Notice Header:**

A header defined for ANNOUNCE method of RFC2326, which is identifying the type of the event pertaining to the ANNOUNCE request.

### 3.1.15 Notice-code

A 4-digit code for the Notice header that identifies the reason for the event pertaining to the ANNOUNCE request.

### 3.1.16 Notice-string

A text string for the Notice header that contains the text description of the reason for the event pertaining to the ANNOUNCE request.

### 3.1.17 Scheduled Transmission Service

A service used to deliver continuous media to a Media Resource Client using specified content transfer parameters (rate, range, etc).

### 3.1.18 Trick Play Media Resource

A Media Resource that contains trick play information used for trick mode operations.

### 3.1.19 Trick Mode

Standard industry terminology for digital media playback at speeds that are slower or faster than normal playback and includes forward and backward directions as well as pause.

### 3.1.20 Uniform Resource Identifier (URI)

A string of characters used to identify a name or a resource. A URI may be either a URL or a URN and it is partitioned into both subspaces [28].

### 3.1.21 Uniform Resource Locator (URL)

A type of URI that specifies the primary access mechanism of a resource by representing its location [27].

### 3.1.22 Uniform Resource Name (URN)

A form of URI that describes a resource using a namespace independent of its location [27].

## 3.2 Identifiers

### 3.2.1 ControlSessionId

The ControlSessionId is used only in the RSTP Proxy case (see sections 6.3-6.7). The ControlSessionId is originated by the IPTV Service Control Function for the RTSP session between the ITF and the IPTV Service Control Function. The ControlSessionId is conveyed to the ITF over the E3 reference point in the 200 OK RTSP SETUP response message. The ITF uses the ControlSessionId to initiate RTSP TEARDOWN.

### 3.2.2 CreatorAssetId

The CreatorAssetId is a globally unique identifier for an asset and is the concatenation of CreatorId and AssetId as follows:

CreatorId / AssetId

The CreatorId is a string that is a registered Internet domain name (e.g., contentOnDemand.com or serviceprovider.com) or a sub-domain of a registered Internet domain name (e.g., abc.contentOnDemand.com or xyz.contentOnDemand.com).

The AssetId is equivalent to the path component of a URI as defined in section 3.3 of RFC 3986. AssetIds are assigned by the entity that is the creator of the asset (as identified by the CreatorId) and must be unique within the realm of the creator.

### 3.2.3 DeviceId

The DeviceId is a globally unique identifier for a device and is the concatenation of IPTVMfgID identifier and MfgAssignedID and *shall* be as defined in ATIS-0800037, *IPTV Device and Authentication Identity Interoperability Specification* [16].

### 3.2.4 MediaSessionId

This MediaSessionId is used only in the RSTP Proxy case (see sections 6.3-6.7). The MediaSessionId is originated by the CD&SF for the RTSP session established between the IPTV Service Control Function and the CD&SF via the S5 reference point. The MediaSessionId is conveyed to the ITF over the E3 reference point in the 200 OK RTSP SETUP response message. The ITF subsequently uses the MediaSessionId to initiate an RTSP PLAY or PAUSE via the E6 reference point.

### 3.2.5 OriginContentId

The OriginContentId is a globally unique identifier for a resource and is the concatenation of OriginId and ContentId as follows:

OriginId/ContentId

The OriginId is equivalent to the host component of an HTTP URI as defined in section 3.2.2 of RFC 2616 [25].

The ContentId is equivalent to the path component of a URI as defined in section 3.3 of RFC 3986. ContentIds are unique within the scope of the OriginId.

### 3.2.6 SessionId

The SessionId is originated by the CD&SF for the RTSP session established between the ITF and CD&SF in the RTSP Redirect case (see sections 6.8-6.11). The ITF subsequently uses the SessionId to initiate an RTSP PLAY, PAUSE and TEARDOWN. The SessionId is also used to identify resources associated with HTTP delivery (see sections 6.12-6.14).

The syntax of the SessionId is that of a "session identifier" as defined in RFC 2326 section 3.4. The length is restricted to a maximum of 255 characters.

### 3.2.7 SubscriberId

The SubscriberId uniquely identifies a person or an organization that enters into a subscription contract with an IPTV service provider. SubscriberId is defined in ATIS-0800037, *IPTV Device and Authentication Identity Interoperability Specification* [16].

### 3.2.8 SuperCasId

The SuperCasId is an identifier that identifies one SCPA from another within the same security system. The SuperCasId is provided to the device by the Service Provider. SuperCasId is defined in ATIS-0800022.v002.

### 3.2.9 UserId

The UserId uniquely identifies an IPTV user (i.e., a person affiliated with an IPTV subscriber: person or organization). UserId is defined in ATIS-0800037, *IPTV Device and Authentication Identity Interoperability Specification* [16].

## 3.3 Acronyms & Abbreviations

| | |
|---|---|
| ATIS | Alliance for Telecommunications Industry Solutions |
| A-UPF | Application User Profile Function |
| CC | Continuity Counter |
| CD&LCF | Content Distribution and Location Control Functions |
| CD&SF | Content Delivery and Storage Functions |
| CoD | Content on Demand |
| CDN | Content Delivery Network |
| DNS | Domain Name System |
| ECM | Entitlement Control Message |
| EMM | Entitlement Management Message |
| EPG | Electronic Programming Guide |
| HD | High Definition |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | HTTP Secure |
| IANA | Internet Assigned Numbers Authority |
| IMS | IP Multimedia Subsystem |

| | |
|---|---|
| ITF | IPTV Terminal Function |
| NGN | Next Generation Network |
| NPT | Normal Play Time |
| PAT | Program Association Table |
| PCR | Program Clock Reference |
| P-CSCF | Proxy-Call Session Control Function |
| PID | Packet Identifier |
| PMT | Program Map Table |
| PPV | Pay Per View |
| PSI | Public Service Identifier |
| PTS | Presentation Time Stamp |
| PVR | Personal Video Recorder |
| RACF | Resource Admission Control Function |
| RCMS | Remote Configuration and Management System |
| RTP | Real-time Transport Protocol |
| RTSP | Real-Time Streaming Protocol |
| SCPA | Service, Content Protection and Authentication |
| S-CSCF | Serving Call Session Control Function |
| SD | Standard Definition |
| SDP | Session Description Protocol |
| SHE | Super Head End |
| SIP | Session Initiation Protocol |
| SP | Service Provider |
| SPS | Sequence Parameter Set |
| S-UPF | Service User Profile Function |
| TLV | Type-Length-Value |
| TS | Transport Stream |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| URN | Uniform Resource Name |
| VHO | Video Hub Office |
| VoD | Video on Demand |
| VSO | Video Serving Office |

# 4 REFERENCE ARCHITECTURE

## 4.1 Introduction

ATIS-0800002, *IPTV Architecture Requirements* [8] defines IPTV as a suite of secure, reliable managed services. CoD is one service from the suite of IPTV services. This section describes the IIF IPTV functional architecture with an emphasis on functions and reference points that are involved with the CoD service.

### 4.2 Functional Architecture

The ATIS High Level Architecture document, ATIS-0800007 [10], describes the architecture for IPTV services, including CoD. Figure 7 of this document has been revised and slightly reformatted as Figure 1 and shows a typical IPTV system, which includes the delivery of various IPTV services including CoD. The architecture supports both non-IMS and IMS-based services in an NGN environment.

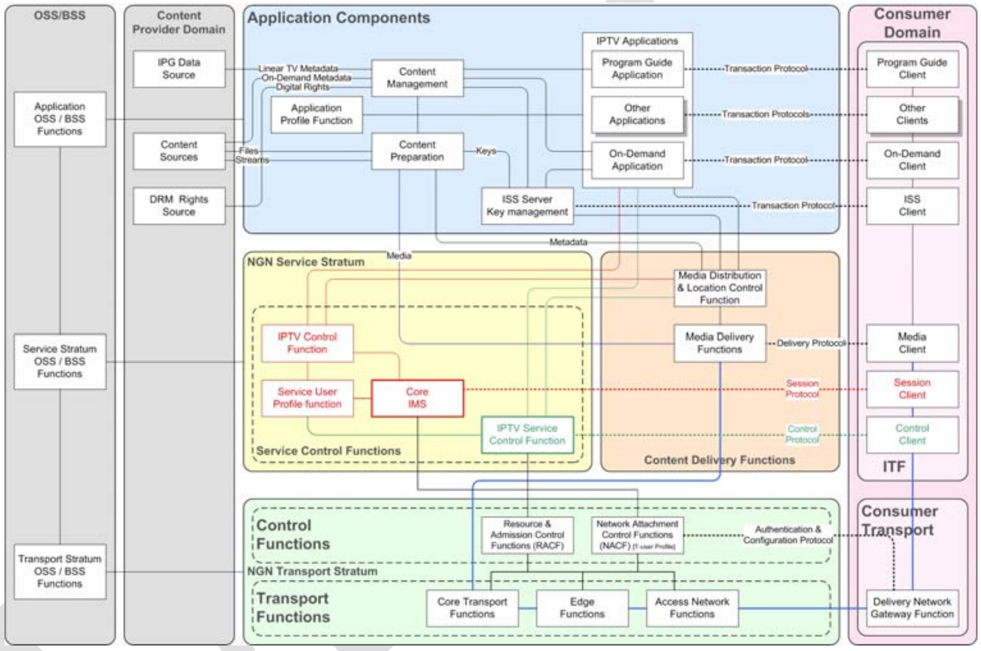


**Figure 1: IPTV High Level Architecture**

Figure 1 highlights the functional entities and interfaces directly involved with the delivery of IPTV services to the ITF (e.g., Linear TV, CoD, etc.). The functions and interfaces that pertain to CoD are described in more detail in the following sections of this specification.

A device implementing the ITF functions for the CoD Service may also implement a number of additional applications for other purposes that are out of scope of the basic CoD operational capabilities. This information flows between the “Other Applications” and “Other Clients” elements of Figure 1.

The separation of the NGN Service Stratum Functions from the content delivery functions in Figure 1 allows these two functions to be provided by different entities within the same service provider domain. For example, a service provider that owns the subscriber relationship by way of the service control functions within the NGN service stratum may choose to provide the content delivery functions by means of a separate entity, such as a third-party content delivery network service provider. In addition, there may be a one-to-many relationship between the service control function and one or more content delivery functions, whereby the service control function can select from among a number of content delivery functions for providing content delivery for a given session.

Figure 2 and Figure 3 show functional architectures for the NGN-based CoD Service. These two figures are the same except for the service control function used and the interfaces involved. Figure 2 uses a non-IMS IPTV service control function. Figure 3 uses an IMS service control function.

To reduce the degree of complexity, Figure 2 and Figure 3 eliminate the following from Figure 1:

- Functions that are part of an IPTV network but out of scope for this document
    - OSS/BSS functions
    - Content provider domain functions
- Functions related to IPTV services other than CoD
    - Linear TV service related functions
    - Interactive Program Guide related functions
    - Other application functions

Figure 2 and Figure 3 provide detail beyond that shown in Figure 1 on functions that are particularly important to the CoD Service. These include:

- Content distribution functions
- Transport resource control
- Asset preparation, including encryption

The thick blue solid line from Asset Preparation Functions, through the Content Delivery and Storage Functions, through the core, edge, access and customer Transport Functions, and on to the ITF represents the physical path of the CoD content. The CoD service typically uses a managed IP network that is not further specified in this document. The dashed lines are logical interfaces. Information exchanged between end-points of the logical interfaces are communicated over the IP transport network along with the CoD content and other IPTV and non-IPTV services that are out of scope. Reference point identifiers are used to identify various interfaces that generally correspond to ITU-T Y.1910 [19] identifiers.

Figure 2 identifies the functions and interfaces involved in the CoD Service using non-IMS IPTV service control. Figure 3 identifies the functions and interfaces involved in the CoD Service with IMS-based service control. These functions and interfaces provide the framework for the flow diagrams in the following sections.

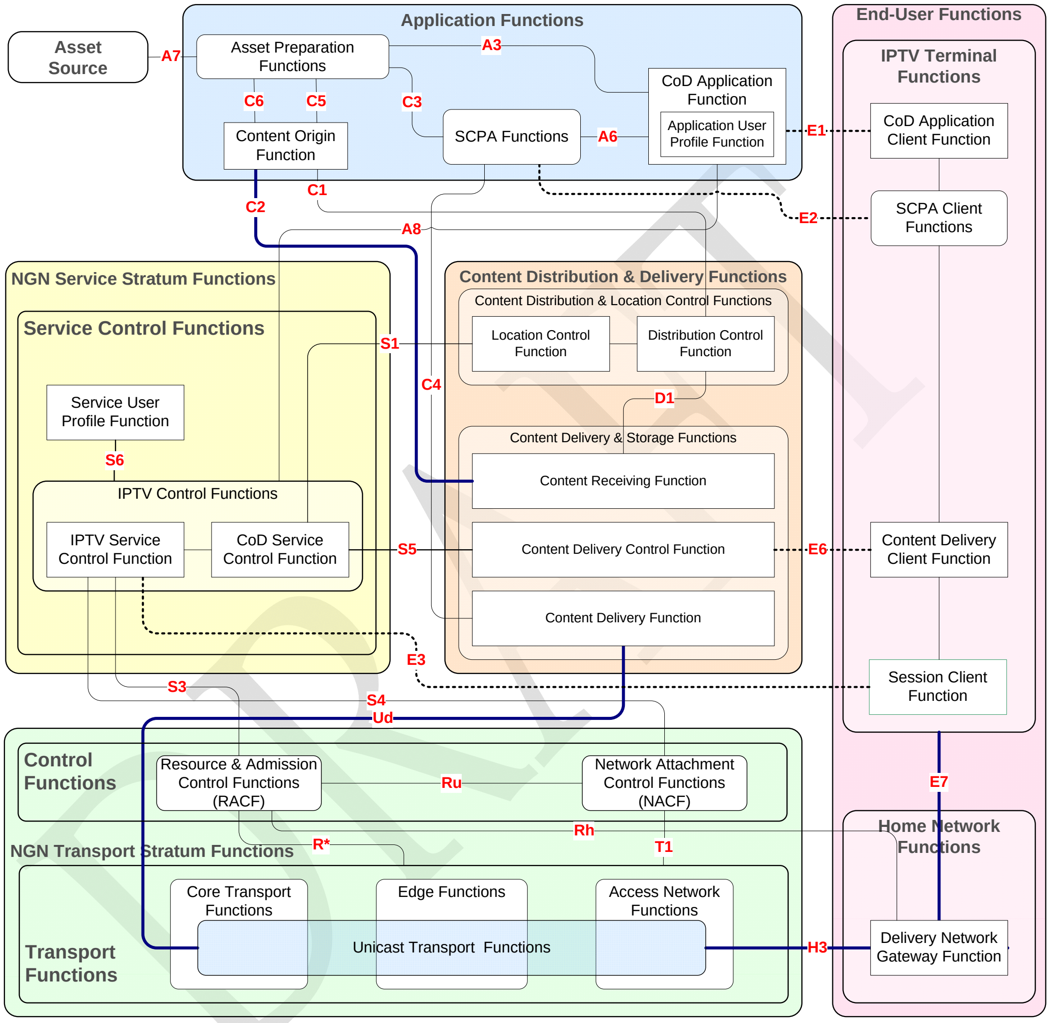

Asset Source
A7
Application Functions
Asset Preparation Functions
A3
C6
C5
C3
CoD Application Function
Application User Profile Function
Content Origin Function
SCPA Functions
A6
E1
End-User Functions
IPTV Terminal Functions
CoD Application Client Function
SCPA Client Functions
E2
C1
C2
A8
NGN Service Stratum Functions
Service Control Functions
Service User Profile Function
S6
IPTV Control Functions
IPTV Service Control Function
CoD Service Control Function
S1
C4
Content Distribution & Delivery Functions
Content Distribution & Location Control Functions
Location Control Function
Distribution Control Function
D1
Content Delivery & Storage Functions
Content Receiving Function
Content Delivery Control Function
Content Delivery Function
S5
E6
Content Delivery Client Function
Session Client Function
E3
S3
S4
Ud
E7
Control Functions
Resource & Admission Control Functions (RACF)
Ru
Network Attachment Control Functions (NACF)
Rh
R*
T1
NGN Transport Stratum Functions
Home Network Functions
Core Transport Functions
Edge Functions
Access Network Functions
Unicast Transport Functions
Transport Functions
H3
Delivery Network Gateway Function

Figure 2: Functional Architecture for non-IMS CoD

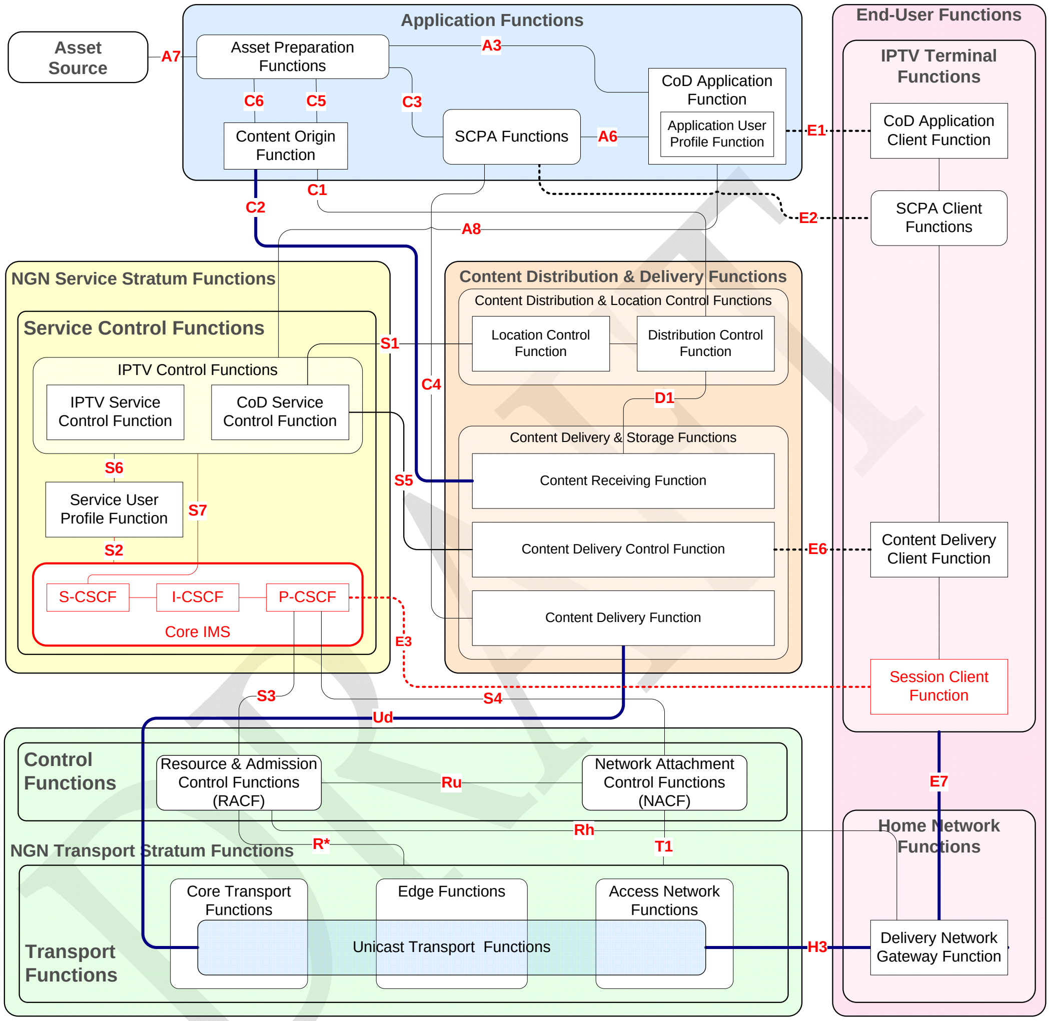

Asset Source
A7
Application Functions
Asset Preparation Functions
A3
C6
C5
C3
CoD Application Function
Application User Profile Function
Content Origin Function
SCPA Functions
A6
E1
C2
C1
A8
E2
End-User Functions
IPTV Terminal Functions
CoD Application Client Function
SCPA Client Functions
NGN Service Stratum Functions
Service Control Functions
IPTV Control Functions
IPTV Service Control Function
CoD Service Control Function
S6
Service User Profile Function
S7
S2
S-CSCF
I-CSCF
P-CSCF
Core IMS
S1
C4
S5
E3
Content Distribution & Delivery Functions
Content Distribution & Location Control Functions
Location Control Function
Distribution Control Function
D1
Content Delivery & Storage Functions
Content Receiving Function
Content Delivery Control Function
Content Delivery Function
E6
Content Delivery Client Function
Session Client Function
S3
S4
Ud
Control Functions
Resource & Admission Control Functions (RACF)
Ru
Network Attachment Control Functions (NACF)
Rh
T1
R*
NGN Transport Stratum Functions
E7
Home Network Functions
Transport Functions
Core Transport Functions
Edge Functions
Access Network Functions
Unicast Transport Functions
H3
Delivery Network Gateway Function

Figure 3: Functional Architecture for IMS CoD

### 4.3 Functional Descriptions

The non-IMS and IMS CoD functional architectures represented in Figure 2 and Figure 3, respectively, are partitioned into five major functional groups:

### 4.3.1 End-User Functions

The End-User Functions include the ITFs and the Home Network Functions. The ITFs enable the end-user to browse, search, select and, where applicable, purchase content. In addition, the ITFs enable the end-user to establish CoD sessions, and provide for the protection and delivery of the selected content. The Home Network Functions provide network connectivity between the service provider access network and the ITF, and are used for the communication of content and control information.

#### 4.3.1.1 SCPA Client Functions

The Service, Content Protection and Authentication (SCPA) Client Functions interact with SCPA Functions to provide service protection, content protection and authentication.

The SCPA Client Functions interacts with the SCPA Functions through at least one reference point. Such reference points are used by the ITF to request and receive security information (e.g., EMMs ATIS-0800001 [7]) from the SCPA Functions.

### 4.3.2 Application Functions

The Application Functions provide the server-side functions necessary for the end-user to browse, search, select and, where applicable, purchase content. Included here are functional elements used in the preparation and storage of content for distribution to Content Delivery and Storage Functions (CD&SF), along with functional elements performing CoD authorization and service control based on user profile and content metadata. Content protection is included in the Application Functions via the distribution of encryption keys to the IPTV terminal, asset preparation, CoD application (pre-encryption) and/or content delivery and storage (session-based encryption) functions.

#### 4.3.2.1 Content Origin Function

The Content Origin Function provides for the storage of content and its eventual delivery to the CD&SF. It is equivalent to the origin server as defined in section 1.3 of RFC 2616 [25]

#### 4.3.2.2 Application User Profile Function

The IPTV Application User Profile Function *may* include:

- End-user settings including information related to the capabilities of the end-user
    - Subscription terms and conditions
    - Parental control level
    - Bandwidth limit
- Global settings
    - Language preference
- User preferences based on CoD metadata (favorites):
    - Genre
    - Artist
    - Format (Black & White, HD, SD, etc)
    - Rating

- Information related to the actions the user may have taken while accessing services, e.g.,:
  - Bookmarks of paused CoD
  - CoD orders and associated status
  - CoD assets that the user has asked to be recorded
- Preference for other services:
  - Linear TV settings
  - Linear TV service packages subscribed to
  - PVR (personal video recorder) settings (PVR preferences network/local, PVR user restrictions, PVR storage limit)
- Subscriber's ITF device capabilities

#### 4.3.2.3 SCPA Functions

The SCPA Functions control and, to a large extent, perform the protection of services and content. Protection includes, for example, control of access to services through authentication and authorization mechanisms and communicating security. While many aspects of SCPA are standardized by IIF or other standard bodies, many other aspects such as conditional access and DRM functions are considered outside the scope of IIF standardization.

### 4.3.3 NGN Service Stratum Functions

The NGN Service Stratum Functions encompass the Service Control Functions for the CoD Service. The Service Control Functions provide service initiation and termination for end-user service requests including service/content authentication against the service user profile, content delivery server location and resource allocation with Content Distribution and Delivery Functions, and network resource reservation with the NGN Transport Stratum Functions. The Service Control Functions encompass the IPTV Control Functions, which is common to all IPTV services (e.g., Linear TV and CoD), and the CoD Service Control Function, which is specific to the CoD Service.

The functional difference between the non-IMS and IMS CoD functional architectures represented by Figures 2 and 3, respectively, are largely manifested in the Service Control Functions. The IMS CoD functional architecture includes the core IMS functional elements that serve to bridge both the IMS-based session client of the ITFs and the IMS-based NGN Transport Stratum Functions with the other functional elements of the CoD functional architecture. More specifically, the Application Functions and Content Distribution and Delivery Functions, along with their respective interfaces, are identical in both non-IMS and IMS functional architectures.

#### 4.3.3.1 Service User Profile Function

There is at least one default user, and there *may* be multiple users associated with a subscriber. Each Service User Profile (S-User Profile) *may* contain:

- user service subscription data (i.e., IPTV services subscribed to)
- subscriber related data (e.g., who pays the incurred charges)
- user location data
- user presence status (e.g., online/offline)
- user profiles for:

- authentication
- authorization
- subscriber mobility
- location
- presence

#### 4.3.3.2 IPTV Service Control Function

IPTV Service Control Function provides the necessary functions for service access authorization to all IPTV services, and validation against service user profiles. It may include the necessary functions for requesting and releasing the network and system resources required to support the related application functionality for some IPTV applications. It may also include the necessary functions to coordinate amongst various IPTV services.

#### 4.3.3.3 CoD Service Control Function

CoD Service Control Function provides the necessary functions to support the CoD Service. This includes session management (establishment, modification and termination) and the tasks associated with that, including resource control management and user service profile enforcement.

### 4.3.4 Content Distribution and Delivery Functions

The Content Distribution and Delivery Functions include the Content Distribution and Location Control Functions (CD&LCF) along with the Content Delivery and Storage Functions (CD&SF). New content available for distribution may be communicated to the CD&LCF by the Content Origin Function within the Application Functions. Included in this communication is the OriginContentId, which includes the location of the Content Origin Function from which the content can be delivered to the appropriate CD&SF. The CD&LCF may control the distribution of available content by providing explicit instructions to the CD&SF to initiate transfer of the available content (push model). Alternatively, content origin location information contained within the OriginContentId, and included in any content request, may be used by the CD&SF to retrieve content from the Content Origin Function on a cache miss (pull model).

On service initiation, the Service Control Functions of the NGN Service Stratum Functions uses the CD&LCF to assist in the selection of an appropriate instance of the CD&SF. The CD&SF selection criteria *may* use location information of the initiating end-user relative to the available servers providing the CD&SF, the protocol used by the ITF for content retrieval (e.g., RTP, HTTP), the availability of the requested content within the CD&SF servers, and/or the current resource load of the CD&SF servers. After selection of an appropriate CD&SF server, the IPTV Service Control Functions forwards or redirects the content request to the selected CD&SF. The CD&SF interacts with the ITFs to control content delivery to the ITF and may provide content protection (i.e., encryption) in cooperation with the Application Functions.

The CD&SF is shown in Figures 2 and 3 as consisting of three internal functional components: Content Receiving Function, Content Delivery Control Function and Content Delivery Function. The Content Receiving Function is used by the CD&SF to obtain content absent from its cache to satisfy an immediate or anticipated request for content. The Content Delivery Control Function is used for session and media control signaling associated with the delivery of content to an ITF. Finally, the Content Delivery Function is used to deliver content to the ITF by way of the NGN Transport Stratum

Functions. As no reference points among these three internal functional components are exposed in this specification, this CD&SF decomposition is provided only to assist the reader in understand the overall CoD functional architecture.

#### 4.3.5 NGN Transport Stratum Functions

The NGN Transport Stratum Functions provide the IP layer connectivity to support all services delivered by IP, including CoD services, to end-users and include Transport Functions and Control Functions. Transport Functions provide the IP layer connectivity between the end-user and each of the application, service control, content delivery and storage, and network provider domains. The Control Functions provide for end-user network attachment and resource allocation for the delivery of the CoD Service in conjunction with the Service Control Functions.

### *4.4 Reference Points Overview*

The reference points identified in Figures 2 and 3 are briefly described below. More detailed specifications for some of these reference points are provided in sections 7 and 10.

#### 4.4.1 Reference Point A3

The A3 reference point is between the Asset Preparation Functions and the COD Application Function. This reference point is used to communicate metadata from the Asset Preparation Functions to the CoD Application Function.

#### 4.4.2 Reference Point A6

The A6 reference point is between the CoD Application Function and the SCPA Functions. The CoD Application Function uses this interface to request content protection information (e.g., ECM). This reference point is for further study.

#### 4.4.3 Reference Point A7

The A7 reference point is between the Asset Source and the Asset Preparation Functions. This reference point is used to ingest content and related metadata into the service provider's system. This reference point is for further study.

#### 4.4.4 Reference Point A8

The A8 reference point is between the CoD Application Function and the IPTV Control Functions. This reference point is used for communicating service information (e.g., NPT, ECM, authorization information) between the CoD Application Function and the IPTV Control Functions. This reference point is used by the IPTV Control Functions to request security information from the CoD Application Function in a pre-encryption environment.

#### 4.4.5 Reference Point C1

The C1 reference point is between the Content Origin Function and the CD&LCF. This reference point is used by the Content Origin Function to notify the CD&LCF of relevant information associated with an asset. This information consists of a distribution policy (i.e., whether an asset needs to be pre-positioned).

#### 4.4.6 Reference Point C2

The C2 reference point is between the Content Origin Function within the Asset Preparation Functions and the Content Receiving Function within the CD&SF. This reference point is used by the Content Receiving Function to retrieve content from the Content Origin Function.

#### 4.4.7 Reference Point C3

The C3 reference point is between the SCPA Functions and the Asset Preparation Functions. This reference point is used for the purpose of exchanging content protection information between the Asset Preparation Functions and the SCPA Functions. This reference point is for further study.

#### 4.4.8 Reference Point C4

The C4 reference point is between the CD&SF and the SCPA Functions. This reference point is used to request security information for the purposes of session-based encryption of the content, if applicable. It is not used for pre-encryption. This reference point is for further study.

#### 4.4.9 Reference Point C5

The C5 reference point is between the Asset Preparation Functions and the Content Origin Function. This reference point is used by the Asset Preparation Functions to notify the Content Origin Function of the availability of an asset (with a named OriginContentId) and its URI on the Asset Preparation Functions.

#### 4.4.10 Reference Point C6

The C6 reference point is between the Asset Preparation Functions and the Content Origin Function. This reference point is used by the Content Origin Function to retrieve content from the Asset Preparation Functions.

#### 4.4.11 Reference Point D1

The D1 reference point is between the CD&LCF and the CD&SF. This reference point is used to communicate content pre-positioned information from the CD&LCF to the CD&SF, as well as for the CD&SF to report status information (e.g., load status, list of stored OriginContentId) to the CD&LCF. The details of this reference point are for further study.

#### 4.4.12 Reference Point E1

The E1 reference point is between the ITF CoD Application Client Function and the CoD Application Function. This reference point is used by the ITF to browse, search, select and, where applicable, purchase content using the services of the CoD Application Function. The catalog, including the CoD metadata elements, is specified in ATIS-0800020.v002 [15]. This reference point is out of scope of this specification.

#### 4.4.13 Reference Point E2

The E2 reference point is between the SCPA Functions and the SCPA Client Function. The SCPA Client Function uses this interface to request content protection information (e.g., EMM) from the SCPA Functions. This reference point is for further study.

#### 4.4.14 Reference Point E3

The E3 reference point is between the ITF Session Client Function and the Service Control Functions. This reference point is used to exchange session signaling information, which includes the OriginContentId of the requested content, between the ITF and the Service Control Functions.

#### 4.4.15 Reference Point E6

The E6 reference point is between the ITF Content Delivery Client Function and the CD&SF. This reference point is used to exchange content control signaling information (e.g., session setup and teardown in the redirect case, play, pause, fast forward, rewind, download) between the ITF and the CD&SF.

#### 4.4.16 Reference Point E7

The E7 reference point between the Delivery Network Gateway Function (DNG) and ITF is used to deliver control messages and content streams. This corresponds to the E7 reference point in ITU-T Y.1910 [21]. This reference point is out of scope of this specification.

#### 4.4.17 Reference Point H3

The H3 reference point between the Unicast Transport Functions and the DNG provides unicast connectivity in order to deliver control messages and content streams. This corresponds to the H3 reference point in ITU-T Y.1910 [21]. This reference point is out of scope of this specification.

#### 4.4.18 Reference Point R*

R* represents all reference points between the RACF and Transport Functions. This *may* include the ITU-T reference points Rc, Rn, Rp, and Rw specified in ITU-T Y.2111 [23], depending on the resource management implementation. These reference points are out of scope of this specification.

#### 4.4.19 Reference Point Ru

The Ru reference point between the NACF and RACF allows the RACF to interact with the NACF for checking on CPE transport subscription profile information and the binding information of the logical/physical port address to an assigned IP address. This corresponds to the Ru reference point in ITU-T Y.2111 [23]. This reference point is out of scope of this specification.

#### 4.4.20 Reference Point Rh

The Rh reference point allows the RACF to push policy decisions to the DNG and also allows the DNG to request admission decisions. ITU-T Y.2111 [23] specifies the functional and information exchange requirements for the Rh reference point. This reference point is out of scope of this specification.

#### 4.4.21 Reference Point S1

The S1 reference point is between the CoD Service Control Function and the Location Control Function within the CD&LCF. This reference point is used by the CoD Service Control Function to locate an instance of the CD&SF capable of delivering the requested content to the ITF.

#### 4.4.22 Reference Point S2

The S2 reference point is between the core IMS and the Service User Profile Function. This reference point *shall* follow the Cx reference point definition as specified by 3GPP TS 23.228 [1], Clause 5.1.2.1 to support the following procedures for information transfer between core IMS and S-UPF:

1) Procedures related to S-CSCF assignment
2) Procedures related to routing information retrieval from S-UPF to core IMS
3) Procedures related to authorization
4) Procedures related to authentication: transfer of security parameters of the subscriber between S-UPF and core IMS
5) Procedures related to transfer of subscription parameters from S-UPF to core IMS

The S2 interface uses the Diameter protocol.

Additional details of the Cx interface can be found in 3GPP TS 29.228 [3] and TS 29.229 [4].

#### 4.4.23 Reference Point S3

The S3 reference point is between the Service Control Functions and RACF. In IMS-based networks, S3 terminates on Core IMS and in non-IMS IPTV networks S3 terminates on the IPTV Control Functions. The Service Control Functions use S3 to request RACF to control transport resources. S3 corresponds to the Rs reference point in ITU-T Y.2111 [23]. The S3 reference point may convey QoS information such as bandwidth via the Max-Requested-Bandwidth-DL AVP and reservation class via the Reservation-Class AVP. The Reservation-Class AVP contains an integer used as an index that identifies a set of traffic characteristics of the flow. This index may point to a configuration of ITU-T Y.1541 [20] parameter values. This specification recommends the use of Y.1541 QoS class 4 for HTTP encapsulated flows delivered over the Ud interface and QoS class 7 for RTP encapsulated flows delivered over the Ud interface. The detailed specification of this reference point is outside the scope of this document.

#### 4.4.24 Reference Point S4

The S4 reference point is between the Service Control Functions and NACF. In IMS-based networks, S4 terminates on Core IMS and in non-IMS IPTV networks S4 terminates on the IPTV Control Functions. The Service Control Functions use S4 to retrieve information related to the IP connectivity (e.g., physical location of the ITF). S4 corresponds to the S-TC1 reference point in ITU-T Y.2014 [22]. This reference point is out of scope.

### 4.4.25 Reference Point S5

The S5 reference point is between the CoD Service Control Function and the CD&SF. This reference point is used to exchange session management information between the CoD Service Control Function and the CD&SF. The S5 reference point corresponds to the non-IMS S5 reference point in ITU-T Y.1910 [21].

### 4.4.26 Reference Point S6

The S6 reference point is between the Service User Profile Function and the IPTV Control Functions. This reference point *shall* follow the Sh reference point definition as specified in 3GPP TS 29.328 [5], Clause 4.2.4a, which is summarized below.

1. The S6 interface is an intra-operator interface.
2. The S6 interface transports transparent data such as service related data and user related information.
3. The S6 interface supports mechanisms for transfer of user related data stored in the S-UPF.
4. The S6 interface supports mechanisms for transfer of standardised data (e.g., for group lists) that can be accessed by IPTV Control Functions.
5. The S6 interface supports mechanisms that allow IPTV Control Functions to activate/deactivate their own existing initial filter criteria stored in the S-UPF on a per subscriber basis.

The S6 interface uses the Diameter protocol.

Additional details of the Sh interface can be found in 3GPP TS 29.328 [5] and TS 29.329 [6].

### 4.4.27 Reference Point S7

The S7 reference point is between the S-CSCF in the core IMS and the IPTV Control Function.

This reference point *shall* follow the ISC reference point definition as specified by 3GPP TS 23.228 [1], clause 4.2.4.

The ISC interface between the S-CSCF and the IPTV Control Functions *shall* be based on SIP and in accordance with the constraints and provisions specified in 3GPP TS 24.229 [2], Annex A.

### 4.4.28 Reference Point T1

The T1 reference point between the NACF and the Access Node's Access Relay Function is used to perform authentication and for the DNG and ITF to obtain necessary information (e.g. IP address, etc.) from the NACF when the DNG and ITF attach to the network. T1 corresponds to the TC-T1 reference point in ITU-T Y.2014 [22] and is based on DHCP as described in ATIS-0800017.v002 [13]. This reference point is out of scope of this specification.

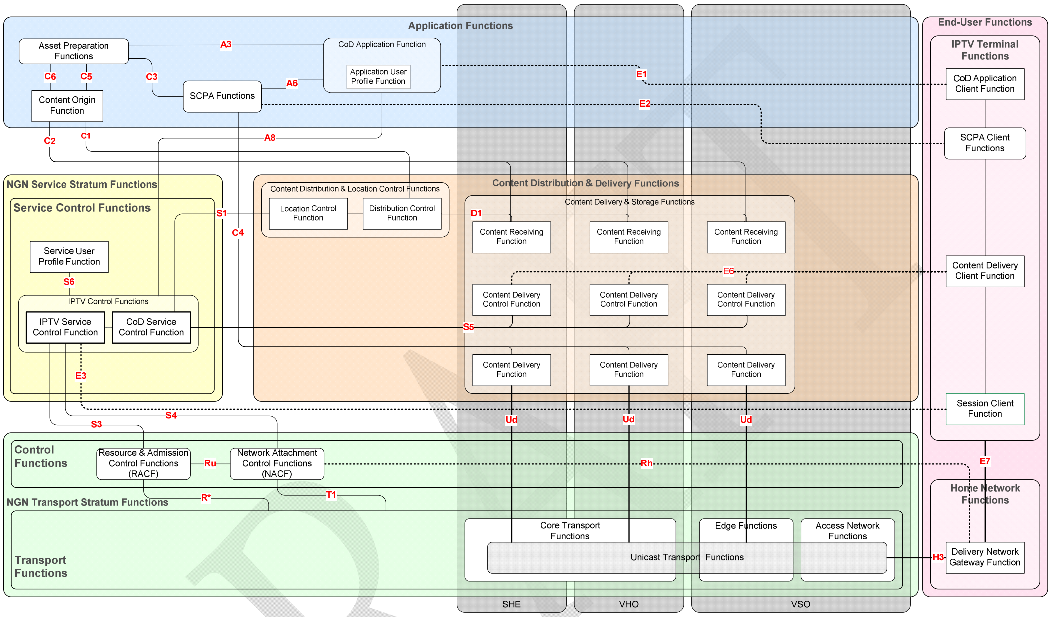
### 4.4.29 Reference Point Ud

The Ud reference point is between the CD&SF and the Unicast Transport Functions. This reference point is used by the CD&SF to deliver content streams in unicast mode.

CoD content can be delivered to ITFs using RTP streaming or HTTP. For HTTP, the response to an E6 HTTP download request is delivered via the Ud reference point. Encoded media formats and constraints are specified in section 9.1 and 9.2. The encoded media *may* be encapsulated in MPEG2 transport streams. MPEG2 transport streams may be encapsulated in RTP or HTTP.

HTTP adaptive streaming is for further study.

## 4.5 *Geographically Distributed Functional Architecture*



Application Functions
Asset Preparation Functions
A3
CoD Application Function
Application User Profile Function
C6
C5
C3
A6
SCPA Functions
Content Origin Function
E1
E2
C2
C1
A8
End-User Functions
IPTV Terminal Functions
CoD Application Client Function
SCPA Client Functions
NGN Service Stratum Functions
Service Control Functions
Service User Profile Function
S6
IPTV Control Functions
IPTV Service Control Function
CoD Service Control Function
S1
C4
Content Distribution & Location Control Functions
Location Control Function
Distribution Control Function
Content Distribution & Delivery Functions
Content Delivery & Storage Functions
D1
Content Receiving Function
Content Receiving Function
Content Receiving Function
E6
Content Delivery Control Function
Content Delivery Control Function
Content Delivery Control Function
Content Delivery Client Function
S5
Content Delivery Function
Content Delivery Function
Content Delivery Function
E3
S4
S3
Ud
Ud
Ud
Session Client Function
Control Functions
Resource & Admission Control Functions (RACF)
Ru
Network Attachment Control Functions (NACF)
Rh
E7
R*
T1
NGN Transport Stratum Functions
Home Network Functions
Core Transport Functions
Edge Functions
Access Network Functions
Unicast Transport Functions
H3
Delivery Network Gateway Function
Transport Functions
SHE
VHO
VSO

**Figure 4: Geographically Distributed Functional Architecture for non-IMS CoD**

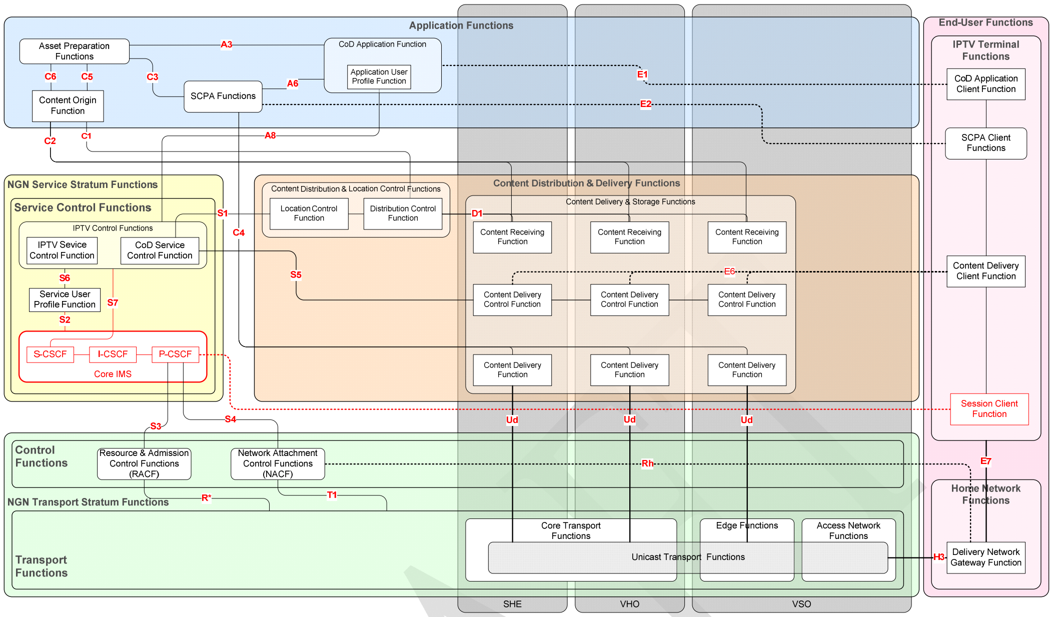

Application Functions
Asset Preparation Functions
A3
CoD Application Function
Application User Profile Function
C6
C5
C3
A6
Content Origin Function
SCPA Functions
E1
E2
C2
C1
A8
End-User Functions
IPTV Terminal Functions
CoD Application Client Function
SCPA Client Functions
NGN Service Stratum Functions
Service Control Functions
IPTV Control Functions
IPTV Sevice Control Function
CoD Service Control Function
S1
C4
Content Distribution & Location Control Functions
Location Control Function
Distribution Control Function
Content Distribution & Delivery Functions
Content Delivery & Storage Functions
D1
Content Receiving Function
Content Receiving Function
Content Receiving Function
S6
S5
E6
Content Delivery Client Function
Service User Profile Function
S7
S2
Content Delivery Control Function
Content Delivery Control Function
Content Delivery Control Function
S-CSCF
I-CSCF
P-CSCF
Core IMS
Content Delivery Function
Content Delivery Function
Content Delivery Function
Session Client Function
S3
S4
Ud
Ud
Ud
Control Functions
Resource & Admission Control Functions (RACF)
Network Attachment Control Functions (NACF)
Rh
E7
NGN Transport Stratum Functions
R*
T1
Home Network Functions
Core Transport Functions
Edge Functions
Access Network Functions
Transport Functions
Unicast Transport Functions
H3
Delivery Network Gateway Function
SHE
VHO
VSO

**Figure 5: Geographically Distributed Functional Architecture for IMS CoD**

The functional elements of the network architecture *may* be mapped to the physical network as shown in Figures 4 and 5 which is a reformatted version of Figure 10 from ATIS-0800007 [10].

This figure includes a network topology as media and control flows from the content provider to the consumer. This topology is intended to be typical, with larger networks having more levels and smaller networks having fewer. Typical network topology nodes are identified as:

- Super Head End (SHE) is the network node with the broadest content scope. The SHE *may* source content to an entire IPTV network and *may* include the primary storage for On-Demand content.
- Video Hub Office (VHO) is the network node with a local/regional content scope. The VHO *may* host region-dependent content (e.g., local programming). It *may* also host more popular content to reduce traffic from the content origin or from a CD&SF in the SHE to the VHO.
- Video Serving Office (VSO) network node connects consumers (via access systems) to the IPTV network. The VSO (typically a Central Office) hosts or connects all access systems for interconnection to consumers. In addition, the VSO *may* contain aggregation equipment to enable interconnection of access systems to the IPTV network. The VSO *may* also host more popular content to reduce traffic from the content origin or from a CD&SF in the SHE/VHO to the VSO.

# 5 Content Distribution and Delivery Overview

## 5.1 Overview of Content Distribution and Control Paths through the Network

Figure 6 provides an overview of the phases of CoD service. Asset preparation is an ongoing process that occurs autonomously from the consumption of CoD service. As preconditions to CoD service, DNG and ITF network attachment, remote configuration, service provider discovery and attachment and services discovery have taken place as specified in ATIS-0800017 [13] and ATIS-0800009 [11]. After the preconditions are satisfied, the CoD service phases are executed. CoD service execution can be described as following a path from content discovery and selection, through one of several alternatives for CoD session management and one of several alternatives of content delivery and control.

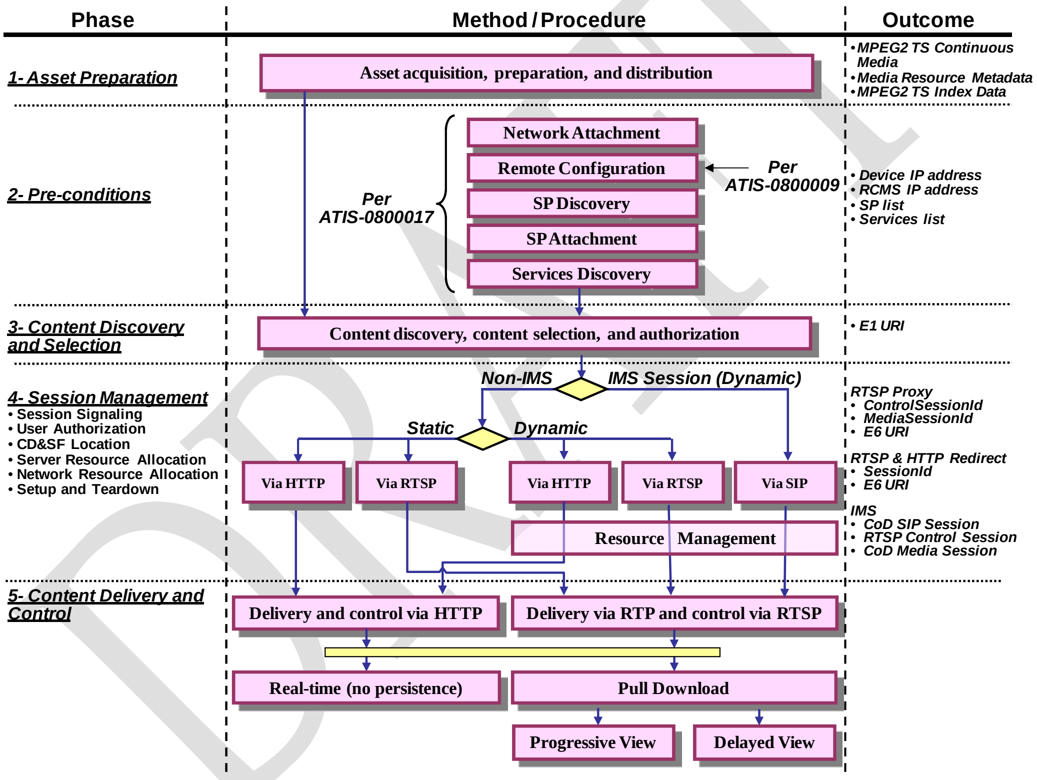


**Figure 6: Overview of CoD Sequence**

The above figure results in the following alternative flows:

**Table 1: Content Delivery Options**

| Resource Allocation Method | Non-IMS/IMS | Session Protocol | Resource Mgmt | Redirect/ Proxy | Delivery/Control | Section Establish/Terminate |
|---|---|---|---|---|---|---|
| Static/Dynamic | Non-IMS | HTTP | No/Yes | Redirect | HTTP/HTTP | 6.12-6.14 |
| Static/Dynamic | Non-IMS | RTSP | No/Yes | Proxy | RTP/RTSP | 6.3/6.5-6.7 |
| Static/Dynamic | Non-IMS | RTSP | No/Yes | Redirect | RTP/RTSP | 6.8/6.9-6.11 |
| Dynamic | IMS Session | SIP | Yes |  | RTP/RTSP | 6.15/6.16-6.18 |

## 5.2 Asset Preparation

CoD content is delivered (via the A7 reference point) in the form of a package that contains content files and metadata, where the metadata describes each content file included in the package. The Asset Preparation Functions prepares the content, generates content metadata and provides for the content life cycle (e.g., selection of content origin and distribution policy). Below is a description of the steps that may be involved in the Asset Preparation Functions:

a. Transcode content
b. Encrypt content
c. Regenerate asset metadata including encryption attributes
d. Generate index data
e. Generate trick play media resources
f. Update asset metadata to indicate index data and trick play media resources.
g. Publish both content and content metadata to the Content Origin Function.
h. Content packaging
i. Content watermarking
j. Ad-insertion, format conversion, resolution conversion, etc

The Asset Preparation Functions announce the availability of the content and its associated Origin URI to the Content Origin Function (via the C5 reference point). The Asset Preparation Functions publish the asset metadata to the CoD Application Function (via the A3 reference point). The catalog is generated and kept updated by the CoD Application Function.

## 5.3 Network and Service Attachment Pre-conditions

The basic IPTV CoD Service operates after the initialization, configuration, service provider discovery, and services discovery documented in ATIS-0800017 [13], *Network Attachment and Initialization of Devices and Client Discovery of IPTV Services* and ATIS-0800009 [11], *Remote Management of Devices in the Consumer Domain for IPTV Services*. Note that those specifications may enable an IPTV Terminal Function (ITF) to access additional IPTV services beyond the basic IPTV CoD Service described in this specification. This specification assumes the ITF is authorized to access one or more service packages.

Mechanisms to restrict or modify the set of available authorized service packages are out of the scope of this specification.

## 5.4 CoD Service Attachment and Initial Resource Acquisition

The CoD Service assumes that an initial set of CoD assets are available through the CoD Metadata prior to the attachment of a user to the service. Subsequent CoD catalog updates and user attachments are asynchronous.

Prior to CoD service attachment, an IPTV network may be engineered and deployed with sufficient network resources available for CoD service and other services (e.g., Linear TV service) such that oversubscription is unlikely to occur no matter the mixture of traffic (unicast/multicast) or expected/subscribed IPTV service penetration. In this case, resource admission control shall be performed for each ITF as it is powered on to authorize the equipment and check the user profile, but network resources are not dynamically allocated (changed) as part of this process by RACF. The network may police these resource limits at the network interface and discard any traffic in excess of these resource limits. A resource reservation function shall be implemented in the application domain to ensure that at any time the total IPTV bandwidth traffic consumed by the household/account on the access link does not exceed the pre-allocated resources. There are two modes of application layer resource management. A CoD application session, which provides association between the service provider and the active ITFs of an IPTV subscriber, is used for this purpose as described in the two sections below.

### 5.4.1 Application Layer with Static Resources

In this case, each CoD client is pre-configured (or is informed at the CoD application session establishment stage) with the maximum bandwidth that it is allowed to use at any point of time on the access link. Note that this bandwidth is allocated to each ITF device individually. The ITF device shall be configurable for remote management access to records of bandwidth consumption.

### 5.4.2 Application Layer Resource Management

In this case, each CoD client is pre-configured (or is informed at the CoD application session establishment stage) with the maximum bandwidth its IPTV household account is allowed to use at any point of time on the access link. Note that in this case, each CoD client shall communicate with other IPTV application clients on the Home Network to ensure that at any time the total of bandwidth (both multicast and unicast) traffic consumed by the household/account on the access link does not exceed the pre-allocated resources.

The protocol for bandwidth reservation and management for clients in the Home Network is out of the scope of this specification.

## 5.5 Authorization Aspects

Initial device authentication occurs during network attachment as defined in ATIS-0800017, using the techniques defined in ATIS-0800037.

The A8, E3 and E6 interfaces use the Authorization header common to HTTP, RTSP and SIP to pass information required for authorization. The Authorization header, as well as the 401 (Unauthorized) response as specified in RFC 2616, includes a scheme and a realm value. The scheme value in a 401

(Unauthorized) response identifies an authentication/authorization scheme that the client must use to obtain service from the server.

RFC 2617 defines two schemes (Basic and Digest) that may be used for authentication/authorization in the context of HTTP, RTSP and SIP. This section defines the ATIS Basic Authorization Scheme specific to ATIS CoD for use on reference points including A8, E3, E6 and S5. This ATIS Basic Authorization Scheme is patterned after the RFC 2617 Basic Scheme.

### 5.5.1 ATIS Basic Authorization Scheme

The ATIS-IIF-BASIC authorization scheme is based on the model that a client must send a set of authorization parameters in each request to a server. The realm value should be considered an opaque string that can only be compared for equality with other realms on that server. The server will service the request only if it can validate the SubscriberId, DeviceId and, optionally, the SuperCasId for the protection space of the Request-URI.

For ATIS-IIF-BASIC, the framework above is utilized as follows:

challenge = " ATIS-IIF-BASIC " realm; realm specified in section 1.1 of RFC 2617

credentials = " ATIS-IIF-BASIC " basic-credentials

Upon receipt of an unauthorized request for a URI within the protection space, the server *may* respond with a challenge such as:

Example challenge:

WWW-Authenticate: ATIS-IIF-BASIC realm=realmname

where "realmname" is the string assigned by the server to identify the protection space of the Request-URI.

To receive authorization, the client sends the SubscriberId, DeviceId and, optionally, SuperCasId, separated by semicolon (";") characters with syntax specified as follows:

basic-credentials = 2*<reqElement ";"> *<optElement>

reqElement = "ATIS-IIF-SubscriberId=" SubscriberId / "ATIS-IIF-DeviceId=" DeviceId

optElement = "ATIS-IIF-SuperCasId=" SuperCasId

For example, if the client wishes to send the SubscriberId "bill" and DeviceId "10:0e:ff:74:6b:90", it would use the following header field:

Example header field:

Authorization: ATIS-IIF-BASIC ATIS-IIF-SubscriberId=bill;ATIS-IIF-DeviceId=10:0e:ff:74:6b:90

A client *should* assume that all paths at or deeper than the depth of the last symbolic element in the path field of the Request-URI are also are within the protection space specified by the ATIS-IIF-BASIC realm value of the current challenge. A client *may* preemptively send the corresponding Authorization header with requests for resources in that space without receipt of another challenge from the server.

# 6 MESSAGE SEQUENCE CHARTS

## 6.1 Asset Preparation and Distribution

Before content can be delivered to a subscriber's ITF, the corresponding asset must be ingested from an Asset Source, prepared by the Asset Preparation Functions, and then the content and its associated metadata (derived from the ingested asset) distributed to different functional components within the Application Functions. Specifically, the content is first published to the Content Origin Function and asset metadata related to the catalog is published to the CoD Application Function.

The distribution of the content to the CD&SF may now follow one of three options:

The first option (A. Non-Real Time Unicast Prepositioning) shown in Figure 7 relates to non-real time prepositioning of the content under the direction of the CD&LCF. This first option uses a unicast method to retrieve content from the Content Origin Function.

The second option (B. Non-Real Time Multicast Prepositioning) shown in Figure 7 relates to non-real time prepositioning of the content using a multicast method. This option is for further study.

The third option (C. Real Time Unicast Retrieval after Cache Miss) shown in Figure 7 represents a real time delivery of content from the Content Origin Function to the CD&SF after a cache miss by the CD&SF. That is, during session setup to a CD&SF, if the CD&SF does not already have the requested content in its storage cache, it has the option of retrieving the content in real time from the Content Origin Function. Here, the progressive retrieval of content from the Content Origin Function may occur at nearly the same time that the content is being delivered to the ITF. It is not a requirement that the CD&SF cache the retrieved content and the precise trigger and timing for content retrieval by the CD&SF is not subject to specification.

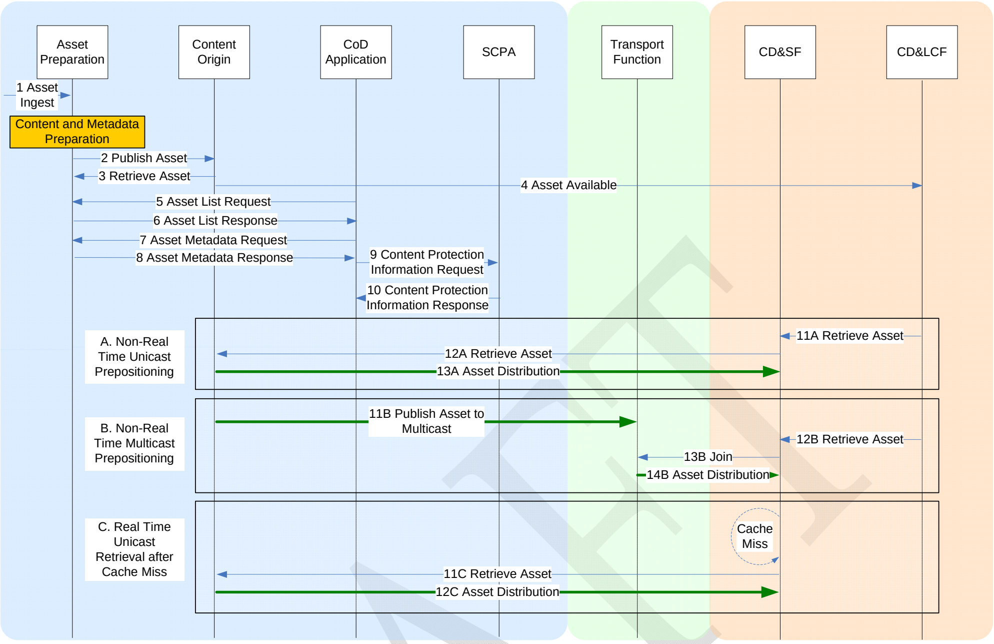


**Figure 7: Asset Preparation and Distribution**

Below is a brief description of the steps in the asset preparation and distribution message flow:

1. Asset ingestion is triggered by receipt of a new asset or other means (A7 create request). The Asset Preparation Functions prepare the asset and generate the OriginContentId and asset metadata, where the asset metadata includes the OriginContentId.
2. The Asset Preparation Functions publish the asset information to the Content Origin Function (C5 create request per Table 8).
3. The Content Origin Function pulls content from the Asset Preparation Functions (C6 download request per Table 5).
4. The Content Origin Function may announce the availability of the asset to the CD&LCF (C1 create request per Table 4).
5. The CoD Application Function sends a query to the Asset Preparation Functions for a list of A3URIs that are of interest (A3 list request per Table 2).
6. The Asset Preparation Functions return a list of A3URIs that satisfy the query (A3 list response per Table 2).
7. The CoD Application Function sends one or more requests for asset metadata (A3 get request per Table 2).
8. The Asset Preparation Functions return the metadata for the requested asset (A3 get response per Table 2).

9. The CoD Application Function requests content protection information (e.g., ECM) for an asset from the SCPA, if encrypted (A6 get request). This step is repeated for all encrypted content.

10. The SCPA returns the associated content protection information (e.g., ECM) to the CoD Application Function (A6 get response). The CoD Application Function may cache the content protection information (e.g., ECM) after it receives it from the SCPA.

Following the previous flow, one of three possible content retrieval flows can then occur:

A. Unicast retrieval of content prior to user content request

11A. The CD&LCF commands the CD&SF to retrieve the content (D1).

12A. The CD&SF requests the content from the Content Origin Function (C2 download request per Table 5).

13A. The Content Origin Function distributes the content to the CD&SF (C2 download response per Table 5).

B. Multicast retrieval of content prior to user content request. This is for further study.

11B. The Content Origin Function publishes the asset to a multicast distribution channel.

12B. The CD&LCF commands the CD&SF to retrieve the content (D1).

13B. The CD&SF joins the multicast distribution channel.

14B. The content is distributed to the CD&SF.

C. Retrieval of content in response to user content request

11C. Upon a cache miss, the CD&SF requests the content from the Content Origin Function (C2 download request per Table 5) using the Origin URI derived from the OriginContentId).

12C. The Content Origin Function distributes the content to the CD&SF (C2 download response per Table 5).

### 6.1.1 Geographically Distributed Content Distribution

Figure 7 may represent a real time communication of content from a different CD&SF location (e.g., VHO - Origin) to the CD&SF (e.g., VSO) after a miss by the CD&SF (e.g., VSO). That is, during session setup to a CD&SF (e.g., VSO), if the CD&SF does not have the requested content in its storage cache, it has the option of retrieving the content in real time from a different CD&SF location (e.g., VHO - Origin).

Within a Geographically Distributed Functional Architecture, there are at least three additional options that *may* be utilized to more effectively scale the network and reduce transport bandwidth.

The first option (D. Real Time Unicast Retrieval after Cache Miss) shown in Figure 8 represents a real time delivery of content from a different CD&SF location (e.g., VHO) to the CD&SF after a cache miss by the CD&SF (e.g., VSO). That is, upon reception of a HTTP GET or RTSP SETUP (redirect) by a CD&SF, if the CD&SF does not already have the requested content in its storage cache, it has the option of retrieving the content in real time from a different CD&SF. This option is almost identical to and *may* be used in combination with the real time delivery shown as option C in Figure 7.

The second option (E. Redirect after Cache Miss) shown in Figure 8 represents a redirect of the client to a different CD&SF location (e.g., VHO) after a cache miss by the CD&SF (e.g., VSO). That is, upon reception of a HTTP GET or RTSP SETUP (redirect) by a CD&SF, if the CD&SF does not have the requested content in its storage cache, it has the option to redirect the ITF to a different CD&SF location that does have or can retrieve the content.

The third option (F. Proxy Redirect after Cache Miss) shown in Figure 8 represents a proxy redirect of the CoD Service Control Function to a different CD&SF location (e.g., VHO) after a cache miss by the CD&SF (e.g., VSO). That is, upon reception of an RTSP SETUP by a CD&SF, if the CD&SF does not have the requested content in its storage cache, it has the option to redirect the CoD Service Control Function to a different CD&SF location that does have or can retrieve the content.

A fourth option (not represented in the below diagram) relates to non-real time prepositioning of the content using a multicast method. This option is for further study.

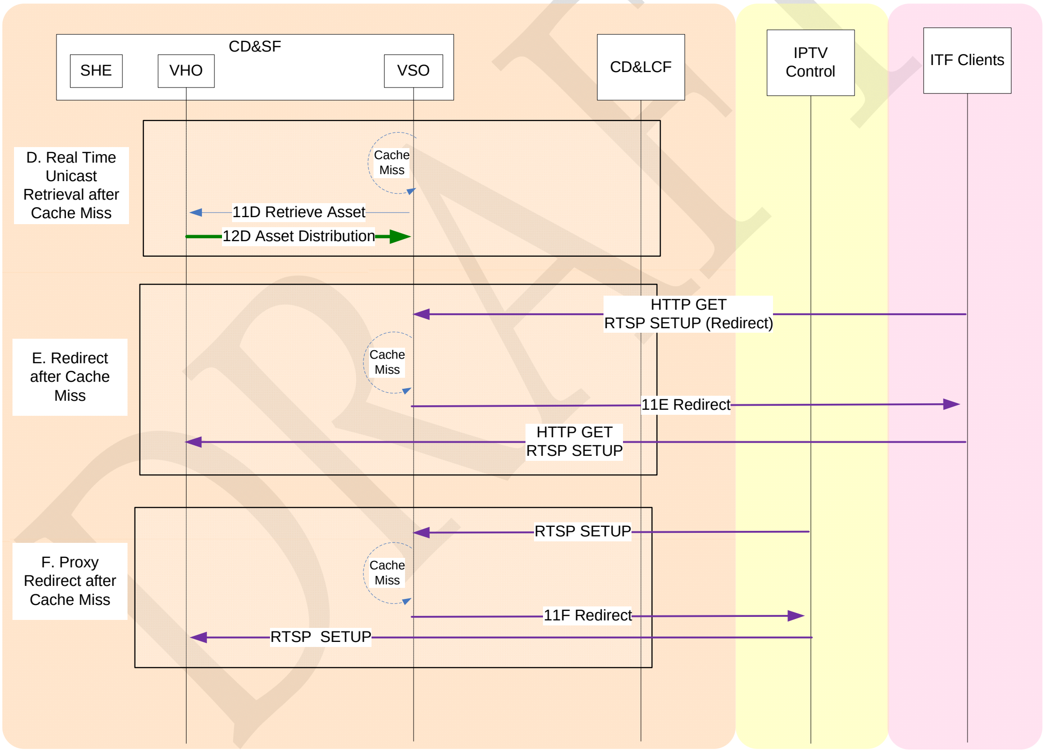


**Figure 8: Geographically Distributed Content Distribution**

D. Alternative retrieval of content in response to user content request

11D. Upon a cache miss, the CD&SF requests the content from the designated alternative CD&SF location (C2 download request per Table 5).

12D. The CD&SF distributes the asset to the CD&SF (C2 download response per Table 5).

E. Redirection to a different content servicing location in response to user content request

11E. Upon a cache miss, the CD&SF responds to the content request of the ITF with a redirect to the designated alternative CD&SF location (E6).

F. Proxy redirection to a  different content servicing location in response to CoD  Service Control session setup request

11F. Upon a cache miss, the CD&SF responds to the session setup request of the CoD Service Control Function with a redirect to the alternate CD&SF location (S5).

### 6.2 Content Selection and Acquisition

The catalog of assets available to the user of the CoD Service is identified through the CoD catalog.  The CoD catalog metadata are specified in *IPTV Electronic Program Guide Metadata Specification*, ATIS 08000020.v002 [15].

The ITF provides a user interface to present this information in a human readable format and permit user manipulation of this information including operations for search, navigation, purchase, authorization and parental controls.  The specification of the user interface is out of scope of this specification.

Catalog and assets for presentation to a specific user *shall* be restricted within the parental control limits associated with that user.

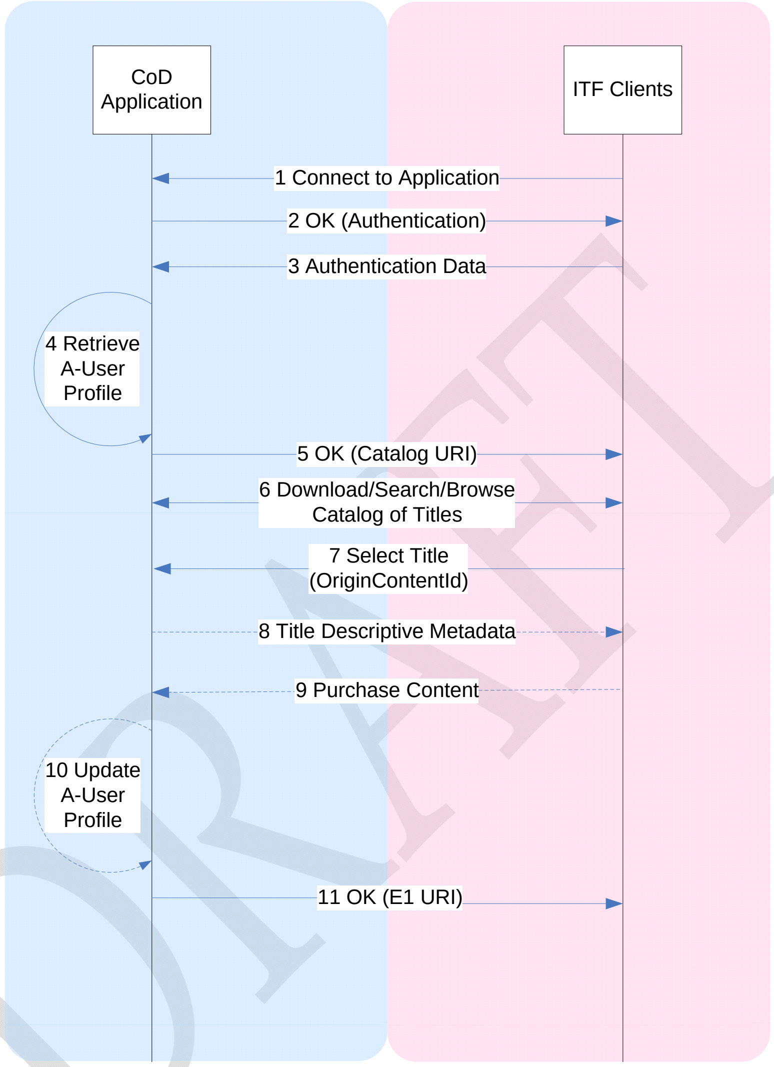


**Figure 9: Content Selection and Acquisition**

As a precondition, network attachment has been achieved through ATIS-0800017 [13]. Below is a brief description of the steps in the content selection and acquisition message flow shown in Figure 9.

1. The ITF connects to the CoD Application Function through user interaction (E1).
2. The CoD Application Function requests data from the ITF identifying the subscriber (E1).
3. The ITF returns data identifying the subscriber for the purposes of authentication to the CoD Application Function (E1).
4. The CoD Application Function validates the subscriber with the A-User Profile.

5. Upon successful authentication, the CoD Application Function returns the URI of the content catalog to the ITF (E1).
6. The ITF accesses the content catalog through one of several possible means and presents the information to the user to allow searching or browsing through the catalog (E1). Refer to ATIS-0800020 [15] for further information on the catalog.
7. The ITF communicates that the user has selected a particular asset identified by an OriginContentId (E1). This message *may* include the Authorization header.
8. The CoD Application Function returns detailed descriptive metadata about the title. If the content requires licensing and/or payment to access and the subscriber has not previously purchased the asset, the CoD Application Function sends asset metadata (including pricing and licensing information, HD/SD, etc.) to the ITF for presentation and agreement by the user (E1). If the asset was previously purchased, the CoD Application Function returns bookmark metadata and this flow ends.
9. The subscriber may purchase the content and the ITF communicates the purchase to the CoD Application Function (E1).
10. The CoD Application Function updates the user's profile on the A-User Profile to record the purchase of the content for later use by the IPTV Control Functions. As a result of this, an association of the SubscriberId and the OriginContentId of the purchased content is maintained by the CoD Application Function.
11. The CoD Application Function indicates to the ITF the successful completion of the purchase and provides the ITF with the E1 URI used for session setup via the E3 reference point.

### *6.3 Non-IMS Proxy CoD Session Establishment – RTSP*

In Figure 10, the RTSP SETUP function in step 2 is proxied to the CD&SF in step 9. A session ID is assigned by the IPTV Control Functions after step 2, which is referred to as the ControlSessionId. An RTSP session ID is created by the CD&SF in response to the proxied RTSP SETUP to the CD&SF in step 9, which is referred to as the MediaSessionId. Both session IDs are then returned in the RTSP 200 OK message of step 13 to the ITF. The IPTV Control Functions may elect to use the value of the MediaSessionId for both. The RTSP PLAY message of step 14 refers to the MediaSessionId. The delivery of the RTSP PLAY message is via a separate TCP session established between the ITF and the CD&SF. The URI of the CD&SF is communicated to the ITF in the RTSP 200 OK message of step 13. Trick play messaging between the ITF and CD&SF is shown in Figure 11. The termination sequence of Figure 12 follows the same path as the RTSP SETUP.

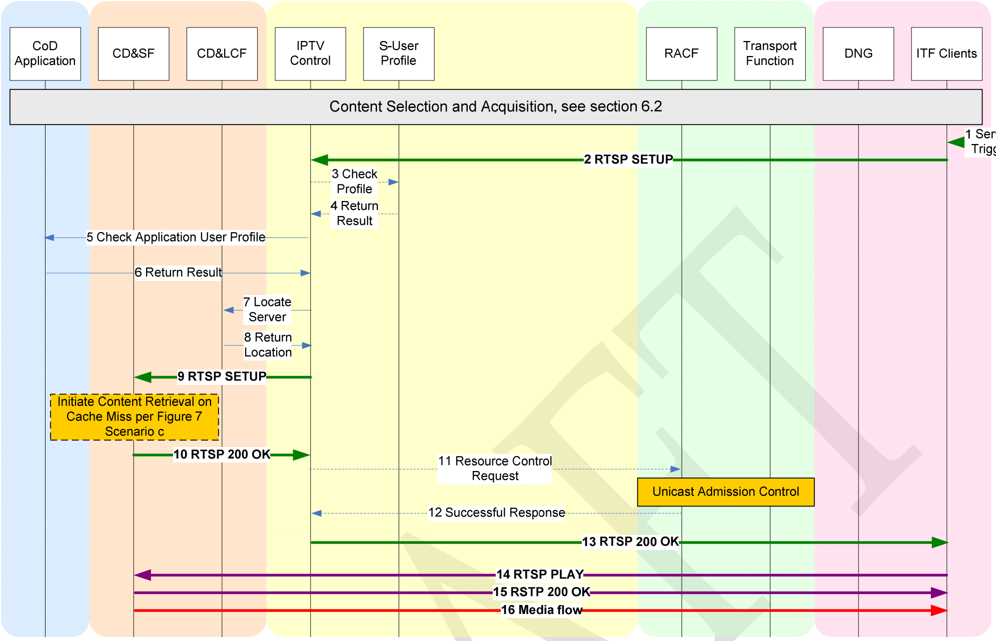


**Figure 10: Non-IMS Proxy CoD Session Establishment - RTSP**

Below is a brief description of the steps in the RTSP-based non-IMS proxy CoD session establishment message flow shown in Figure 10~~Figure 10~~.

1. The CoD Service is triggered through user interaction, preset recording instructions or other means.
2. The ITF initiates an RTSP control session by sending an RTSP SETUP to the IPTV Control Function, including the OriginContentId provided by the CoD Application Function when the content was selected (E3 setup request per Table 10).
3. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
4. The S-User Profile returns the profile data which the IPTV Control Functions uses to validate the service connection (S6) (optional).
5. The IPTV Control Functions request the subscriber's authorization for delivery of the asset (A8 authorization request per Table 3).
6. If the subscriber is authorized, the Application Function returns the allocated session bandwidth and, optionally, content protection information (e.g., ECM) and NPT value (A8 authorization response per Table 3).
7. The IPTV Control Functions request the location of a server within the CD&SF for content delivery from the CD&LCF (S1 locate request per Table 21).
8. The CD&LCF returns the location (CD&SF host) of the chosen server within the CD&SF to the IPTV Control Functions (S1 locate response per Table 21).
9. The IPTV Control Functions initiate an RTSP media session by sending an RTSP SETUP to the specified server location in the CD&SF (S5 setup request per Table 22).

10. The CD&SF establishes the RTSP media session and returns an RTSP 200 OK to the IPTV Control Functions including the MediaSessionId (S5 setup response per Table 22).
11. The IPTV Control Functions initiate a resource reservation request to the RACF based on the bandwidth for the requested CoD service stream. The RACF allocates the required bandwidth for the service (S3) (optional).
12. The RACF signals that the bandwidth allocation is complete to the IPTV Control Functions (S3) (optional).
13. The IPTV Control Functions return an RTSP 200 OK to the ITF, including the ControlSessionId, MediaSessionId and location (E6 URI) of the server within the CD&SF (E3 setup response per Table 10).
14. The ITF sends an RTSP PLAY for the signaled MediaSessionId to the location of the server within the CD&SF (E6 play request per Table 15).
15. The CD&SF starts media playback within the RTSP media session and returns an RTSP 200 OK to the ITF (E6 play response per Table 15).
16. The media is now flowing to the ITF (Ud).

### 6.4 RTSP Trick Play

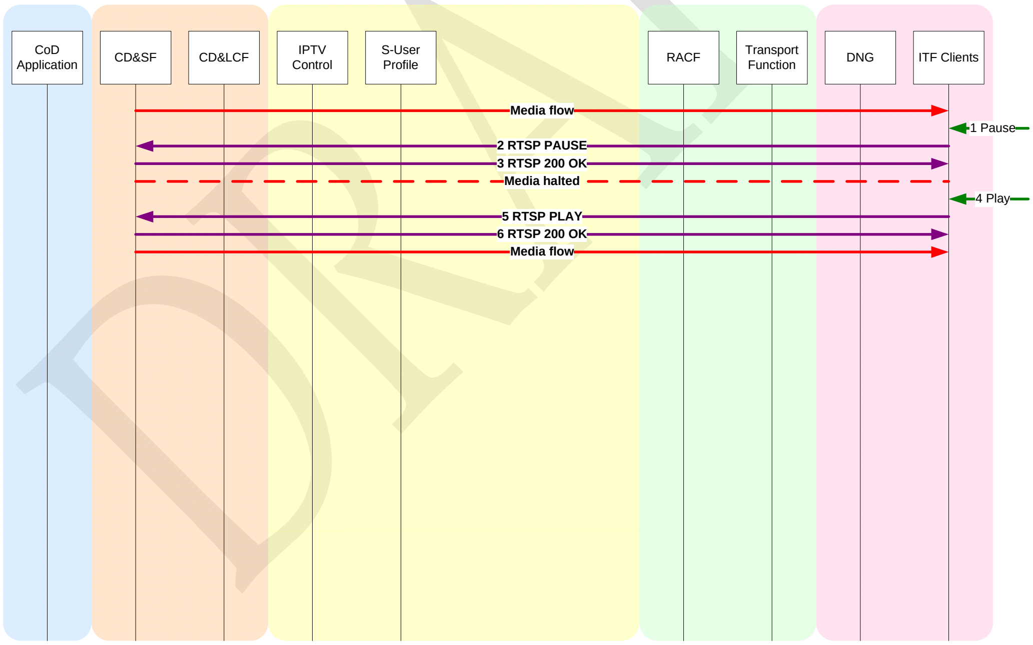


**Figure 11: RTSP Trick Play**

Below is a brief description of the steps in the RTSP trick play message flow shown in Figure 11.

1. A pause of the media playout is triggered through user interaction.

2. The ITF sends an RTSP PAUSE to the CD&SF (E6 pause request per Table 15).
3. The CD&SF pauses the media flow within the RTSP session and returns an RTSP 200 OK to the ITF (E6 pause response per Table 15). The media flow is now halted.
4. A resumption of the media playout is triggered by the user.
5. The ITF sends an RTSP PLAY to the CD&SF (E6 play request per Table 15).
6. The CD&SF resumes the media flow within the RTSP session and returns an RTSP 200 OK to the ITF (E6 play response per Table 15). The media is now flowing to the ITF.

## *6.5 Non-IMS Proxy CoD Session Termination – RTSP*

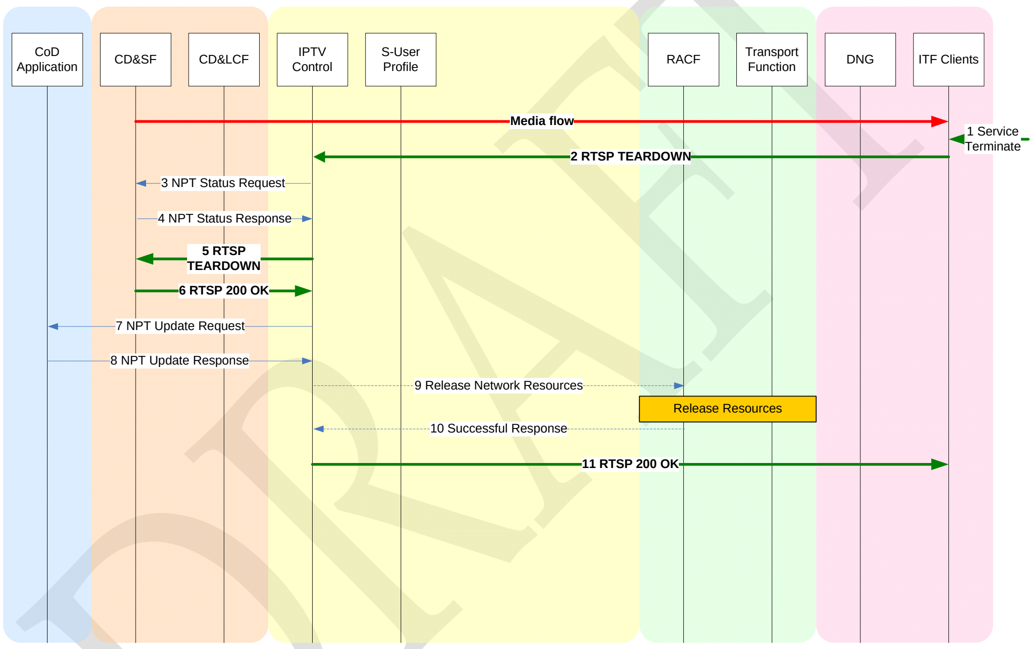


**Figure 12: Non-IMS Proxy CoD Session Termination - RTSP**

Below is a brief description of the steps in the RTSP-based non-IMS proxy CoD session termination message flow shown in Figure 12~~Figure 12~~.

1. The CoD Service termination is triggered through user interaction or other means.
2. The ITF sends an RTSP TEARDOWN to the IPTV Control Functions (E3 teardown request per Table 10).
3. The IPTV Control Functions request the NPT value from the CD&SF (S5 status request per Table 22).
4. The CD&SF returns the NPT value (S5 status response per Table 22).
5. The IPTV Control Functions forwards the RTSP TEARDOWN to the CD&SF (S5 teardown request per Table 22).

6. The CD&SF tears down the RTSP session and returns an RTSP 200 OK to the IPTV Control Functions (S5 teardown response per Table 22).
7. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
8. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
9. The IPTV Control Functions request the RACF to deallocate the network resources for the CoD Service in the transport network (S3) (optional). The RACF deallocates the bandwidth needed by the transport function.
10. The RACF signals that the bandwidth deallocation is complete to the IPTV Control Functions (S3) (optional).
11. The IPTV Control Functions send RTSP 200 OK to the ITF (E3 teardown response per Table 10). The service is now terminated.

## *6.6 Non-IMS Proxy CD&SF Initiated Session Termination – RTSP*

In addition to the initiation of a session teardown from an ITF, there may be cases where the CD&SF needs to terminate an RTSP session. This requires an asynchronous message, which references the MediaSessionId, from the CD&SF to the IPTV Control. The reasons for this capability being necessary may include such conditions as the detection of error conditions, manually initiated events or service provider business rules.

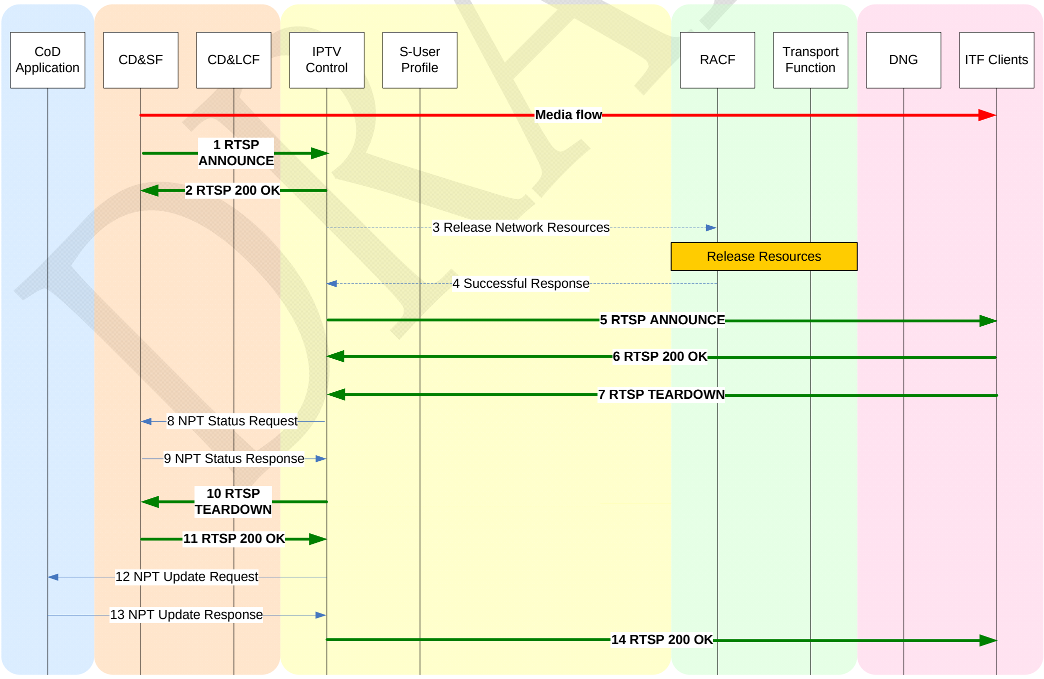


**Figure 13: Non-IMS Proxy CD&SF Initiated Session Termination – RTSP**

As a pre-requisite it is assumed that the media is flowing to the ITF and there is an RTSP control session in place between the ITF and the IPTV Control Functions and an RTSP media session between the ITF and the CD&SF.

Below is a brief description of the steps in the RTSP-based non-IMS proxy CD&SF initiated session termination message flow shown in Figure 13.

1. The CD&SF disconnects the RTSP media session locally and sends an RTSP ANNOUNCE to the IPTV Control Functions to announce that the RTSP media session has been terminated and the reason for termination (S5 event request per Table 22).
2. The IPTV Control Functions acknowledge the RTSP ANNOUNCE and return an RTSP 200 OK to the CD&SF (S5 event response per Table 22).
3. The IPTV Control Functions release the network resources associated with the CoD session (S3) (optional).
4. The RACF acknowledges release of the resources (S3) (optional).
5. The IPTV Control Functions send an RTSP ANNOUNCE to the ITF client to announce that the RTSP control session has been terminated and the reason for termination (E3 event request per Table 10).
6. The ITF client acknowledges the RTSP ANNOUNCE and returns an RTSP 200 OK to the IPTV Control Functions (E3 event response per Table 10).
7. The ITF sends an RTSP TEARDOWN to the IPTV Control Functions for the RTSP control session (E3 teardown request per Table 10).
8. The IPTV Control Functions request the NPT value from the CD&SF (S5 status request per Table 22).
9. The CD&SF returns the NPT value (S5 status response per Table 22).
10. The IPTV Control Functions initiate an RTSP TEARDOWN to the CD&SF for the RTSP media session (S5 teardown request per Table 22).
11. The CD&SF tears down the RTSP media session and returns an RTSP 200 OK to the IPTV Control Functions (S5 teardown response per Table 22).
12. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
13. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
14. The IPTV Service Control Functions send an RTSP 200 OK to the ITF (E3 teardown response per Table 10). The service is now terminated.

## 6.7 Non-IMS Proxy CoD Service Control Initiated Session Termination – RTSP

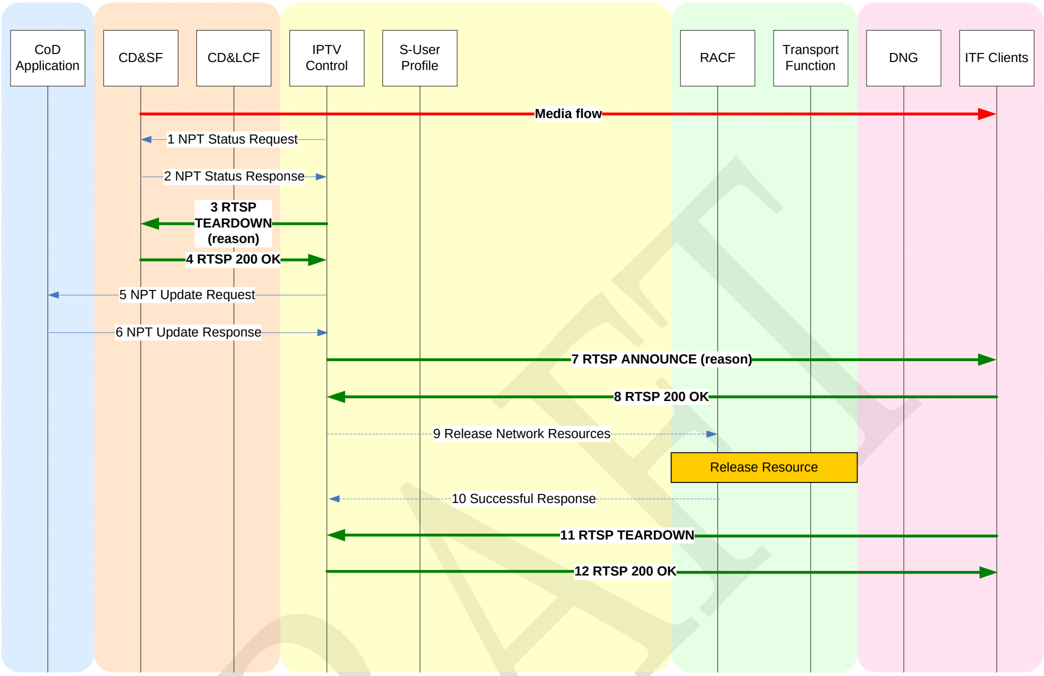


**Figure 14: Non-IMS Proxy CoD Service Control Initiated Session Termination – RTSP**

As a pre-requisite it is assumed that the media is flowing to the ITF and there is an RTSP session in place between the ITF and the IPTV Control Functions.

Below is a brief description of the steps in the RTSP-based non-IMS proxy IPTV Control initiated session termination message flow shown in Figure 14.

1. The IPTV Control Functions request the NPT value from the CD&SF (S5 status request per Table 22).
2. The CD&SF returns the NPT value (S5 status response per Table 22).
3. The IPTV Control Functions send an RTSP TEARDOWN to the CD&SF to announce that the RTSP has been terminated and the reason for termination (S5 teardown request per Table 22).
4. The CD&SF acknowledges the RTSP TEARDOWN and returns an RTSP 200 OK to the IPTV Control Functions (S5 teardown response per Table 22).
5. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
6. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
7. The IPTV Control Functions send an RTSP ANNOUNCE to the ITF client to announce that the RTSP has been terminated and the reason for termination (E3 event request per Table 10).

8. The ITF client acknowledges the RTSP ANNOUNCE and returns an RTSP 200 OK to the IPTV Control Functions (E3 event response per Table 10).
9. The IPTV Control Functions request the RACF to release the network resources associated with the session (S3) (optional).
10. The RACF acknowledges release of the resources (S3) (optional).
11. The ITF sends an RTSP TEARDOWN to the IPTV Control Functions (E3 teardown request per Table 10).
12. The IPTV Control Functions acknowledge the RTSP TEARDOWN and return an RTSP 200 OK to the ITF (E3 teardown response per Table 10). The service is now terminated.

### 6.8 Non-IMS Redirect CoD Session Establishment – RTSP

In Figure 15, the RTSP SETUP in step 2 followed by the RTSP REDIRECT in step 7 is used only to locate a server within the CD&SF and communicate the result to the ITF. No resources are allocated at this point and no RTSP session is established. The RTSP SETUP of step 8 is directly from the ITF to the CD&SF and initiates the sequence of resource allocations. Following the RTSP 200 OK of step 15, an RTSP session is established between the ITF and CD&SF. The termination sequence of Figure 16 follows the same path as the RTSP SETUP.

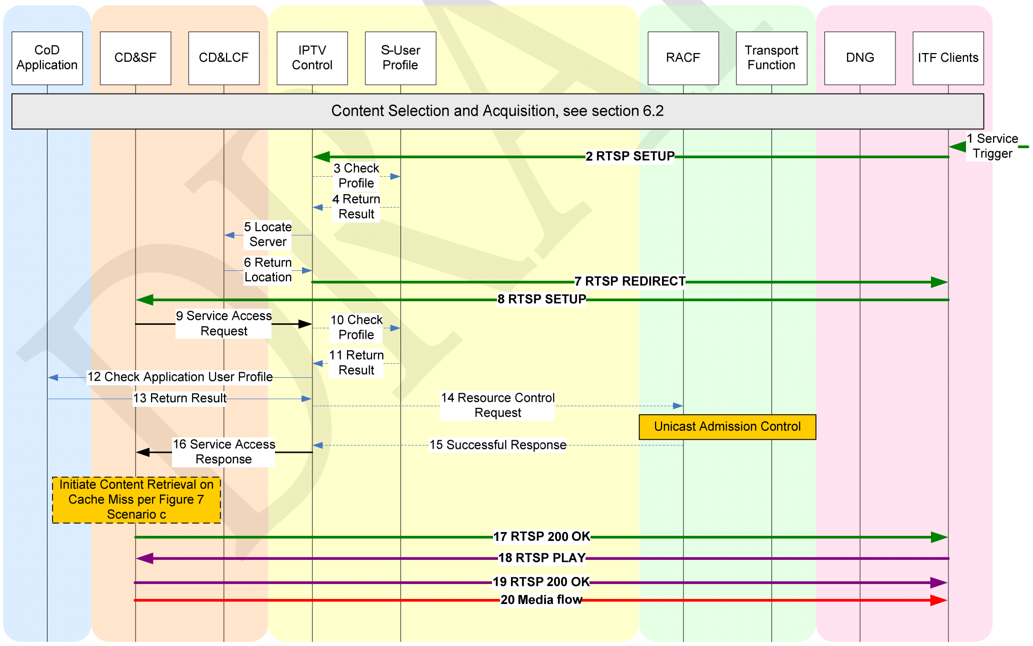


**Figure 15: Non-IMS Redirect CoD Session Establishment – RTSP**

Below is a brief description of the steps in the RTSP-based non-IMS redirect CoD session establishment message flow shown in Figure 15~~Figure 15~~.

1. The CoD Service is triggered through user interaction, preset recording instructions or other means.
2. The ITF initiates an RTSP session by sending an RTSP SETUP to the IPTV Control Functions, including the OriginContentId provided by the CoD Application Function when the content was selected (E3 setup request per Table 11).
3. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
4. The S-User Profile returns the profile data which the IPTV Control Functions use to validate the service connection (S6) (optional).
5. The IPTV Control Functions request the location of a server within the CD&SF for content delivery from the CD&LCF (S1 locate request per Table 21).
6. The CD&LCF returns the location of the chosen server (CD&SF URI) within the CD&SF to the IPTV Control Functions (S1 locate response per Table 21).
7. The IPTV Control Functions return an RTSP REDIRECT to the ITF containing the location of the chosen server (CD&SF host) within the CD&SF (E3 setup response per Table 11).
8. The ITF sends an RTSP SETUP to the specified location in the CD&SF (E6 setup request per Table 16).
9. The CD&SF informs the IPTV Control Functions that a request for CoD Service has been made by the ITF (S5 access request per Table 25).
10. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
11. The S-User Profile returns the profile data which the IPTV Control Functions use to validate the service connection (S6) (optional).
12. The IPTV Control Functions request the subscriber's authorization for delivery of the asset (A8 authorization request per Table 3).
13. If the subscriber is authorized, the CoD Application Function returns the allocated session bandwidth and, optionally, content protection information (e.g., ECM) and NPT value (A8 authorization response per Table 3).
14. The IPTV Control Functions initiate a resource reservation request to the RACF based on the bandwidth for the requested IPTV stream (S3) (optional). The RACF allocates the required bandwidth for the service.
15. The RACF signals that the bandwidth allocation is complete to the IPTV Control Functions (S3) (optional).
16. The IPTV Control Functions signal the approval of the CoD Service to the CD&SF (S5 access response per Table 25).
17. The CD&SF returns RTSP 200 OK to the ITF (E6 setup response per Table 16).
18. The ITF sends an RTSP PLAY to the location of the media server within the CD&SF (E6 play request per Table 16).
19. The CD&SF starts media playback within the RTSP session and returns 200 OK to the ITF (E6 play response per Table 16).
20. The media is now flowing to the ITF (Ud).

## 6.9 Non-IMS Redirect CoD Session Termination - RTSP

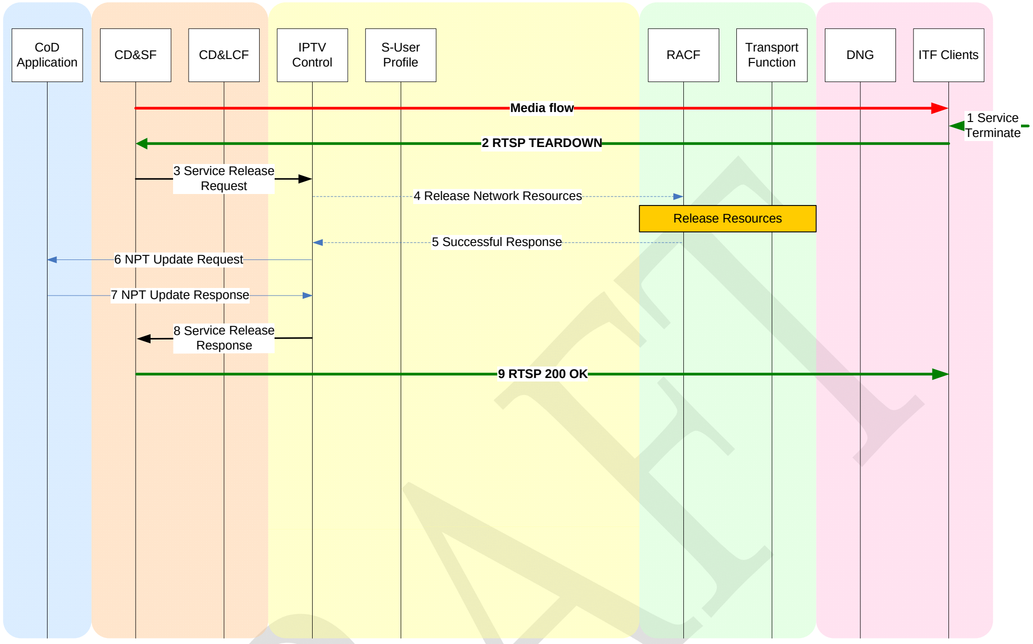


**Figure 16: Non-IMS Redirect CoD Session Termination - RTSP**

Below is a brief description of the steps in the RTSP-based non-IMS redirect CoD session termination message flow shown in Figure 16.

1. The CoD service termination is triggered through user interaction or other means.
2. The ITF sends an RTSP TEARDOWN to the CD&SF (E6 teardown request per Table 16).
3. The CD&SF tears down the RTSP session and requests that the CoD Service session be released by the IPTV Control Functions (S5 release request per Table 25).
4. The IPTV Control Functions request the RACF to deallocate the network resources for the CoD Service in the transport network (S3) (optional). The RACF deallocates the bandwidth needed by the transport function.
5. The RACF signals that the bandwidth deallocation is complete to the IPTV Control Functions (S3) (optional).
6. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
7. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
8. The IPTV Control Functions inform the CD&SF that the session has been released (S5 release response per Table 25).
9. The CD&SF sends RTSP 200 OK to the ITF (E6 teardown response per Table 16). The service is now terminated.

## 6.10 Non-IMS Redirect CD&SF Initiated Session Termination - RTSP

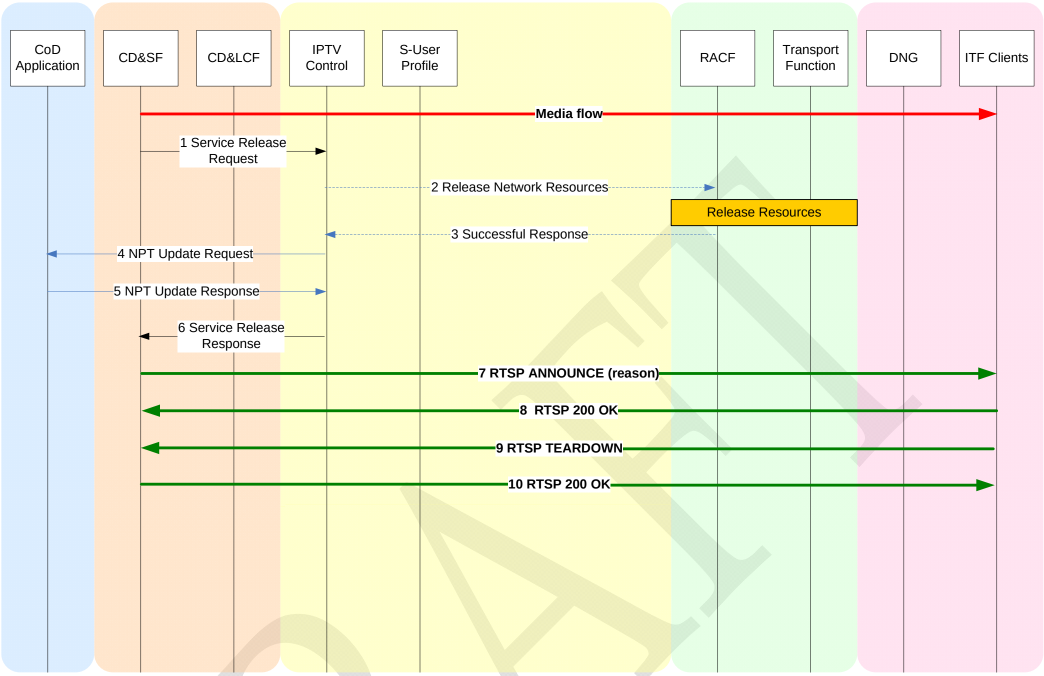


**Figure 17: Non-IMS Redirect CD&SF Initiated Session Termination - RTSP**

Below is a brief description of the steps in the RTSP-based non-IMS redirect CD&SF initiated session termination message flow shown in Figure 17. Note: Steps 1 and 5 may be simultaneous. Also, as a pre-requisite, it is assumed that the media is flowing to the ITF and there is an RTSP session in place between the ITF and the CD&SF.

1. The CD&SF disconnects locally the RTSP session and sends a Service Release Request to the IPTV Control Functions to announce that the RTSP has been terminated and the reason for termination (S5 release request per Table 25).
2. The IPTV Control Functions release the network resources associated with the session (S3) (optional).
3. The RACF acknowledges release of the resources (S3) (optional).
4. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
5. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
6. The IPTV Control Functions send a Service Release Response to the CD&SF (S5 release response per Table 25).
7. The CD&SF sends an RTSP ANNOUNCE to the ITF client to announce that the RTSP has been terminated and the reason for termination (E6 event request per Table 16).

8. The ITF client acknowledges the RTSP ANNOUNCE and returns an RTSP 200 OK to the CD&SF (E6 event response per Table 16).
9. The ITF sends an RTSP TEARDOWN to the CD&SF (E6 teardown request per Table 16).
10. The CD&SF acknowledges the RTSP TEARDOWN and returns an RTSP 200 OK to the ITF (E6 teardown response per Table 16). The service is now terminated.

### *6.11 Non-IMS Redirect IPTV Control Functions Initiated Termination – RTSP*

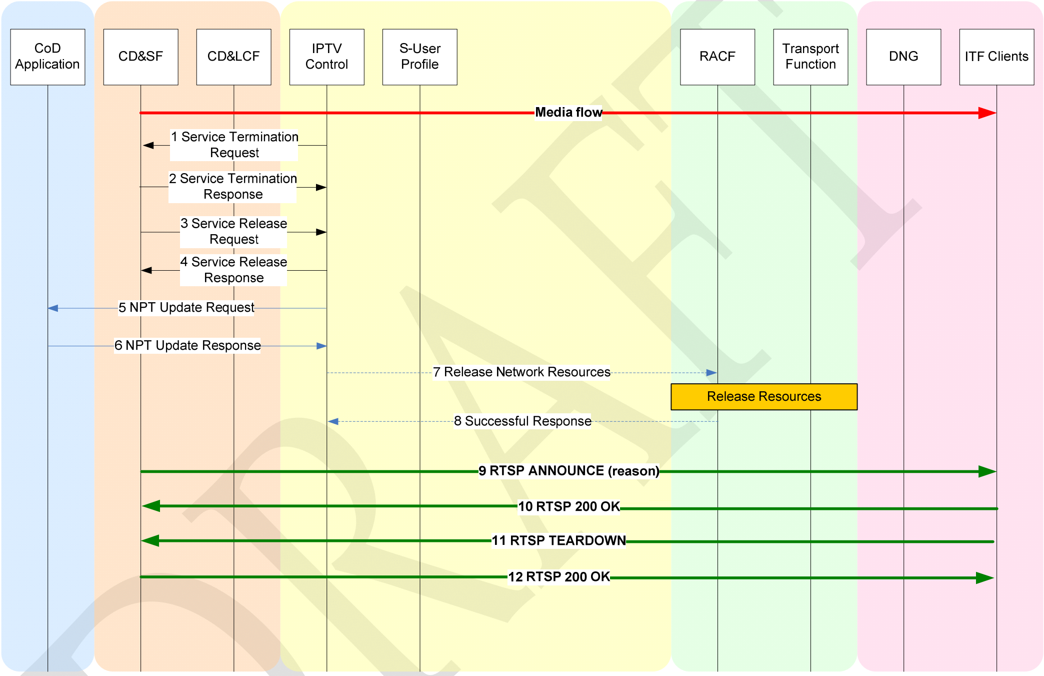


**Figure 18: Non-IMS Redirect IPTV Control Functions Initiated Termination – RTSP**

Below is a brief description of the steps in the RTSP-based non-IMS redirect IPTV Control initiated termination message flow shown in Figure 18. As a pre-requisite it is assumed that the media is flowing to the ITF and there is an RTSP session in place between the ITF and the CD&SF.

1. The IPTV Control Functions send a Service Termination Request to the CD&SF to inform the CD&SF that the RTSP session must be torn down (S5 terminate request per Table 25).
2. The CD&SF acknowledges the Service Termination Request and returns a Service Termination Response to the IPTV Control Functions (S5 terminate response per Table 25).
3. The CD&SF sends a service release request to the IPTV Control Functions, which includes the NPT value (S5 release request per Table 25).
4. The IPTV Control Functions acknowledge the Service Release Request (S5 release response per Table 25).

5. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
6. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
7. The IPTV Control Functions request the RACF to release the network resources associated with the session (S3) (optional).
8. The RACF acknowledges release of the resources (S3) (optional).
9. The CD&SF sends an RTSP ANNOUNCE to the ITF client to announce that the RTSP has been terminated and the reason for termination (E6 event request per Table 16).
10. The ITF client acknowledges the RTSP ANNOUNCE and returns an RTSP 200 OK to the CD&SF (E6 event response per Table 16).
11. The ITF sends an RTSP TEARDOWN to the CD&SF (E6 teardown request per Table 16).
12. The CD&SF acknowledges the RTSP TEARDOWN and returns an RTSP 200 OK to the ITF (E6 teardown response per Table 16). The service is now terminated.

### 6.12 Redirect CoD Session – HTTP Content Download

Figure 19 illustrates session management using HTTP. In the illustrated use case, the ITF does not explicitly signal the beginning and end of the HTTP session via the E3/E6 POST and E6 DELETE methods. Instead, the ITF requests the delivery of a resource using a single HTTP GET transaction. The HTTP GET in step 2 followed by the HTTP 302 REDIRECT in step 7 are used to locate a server within the CD&SF and communicate the result to the ITF. No resources are allocated at this point and no state of the request is maintained. The HTTP GET of step 8, directly from the ITF to the chosen server within the CD&SF, initiates the sequence of resource allocations and results in the HTTP 200 OK of step 17. The HTTP 200 OK of step 17 includes the delivery of the requested content. When the delivery of the requested content is complete, one can consider the HTTP session terminated.

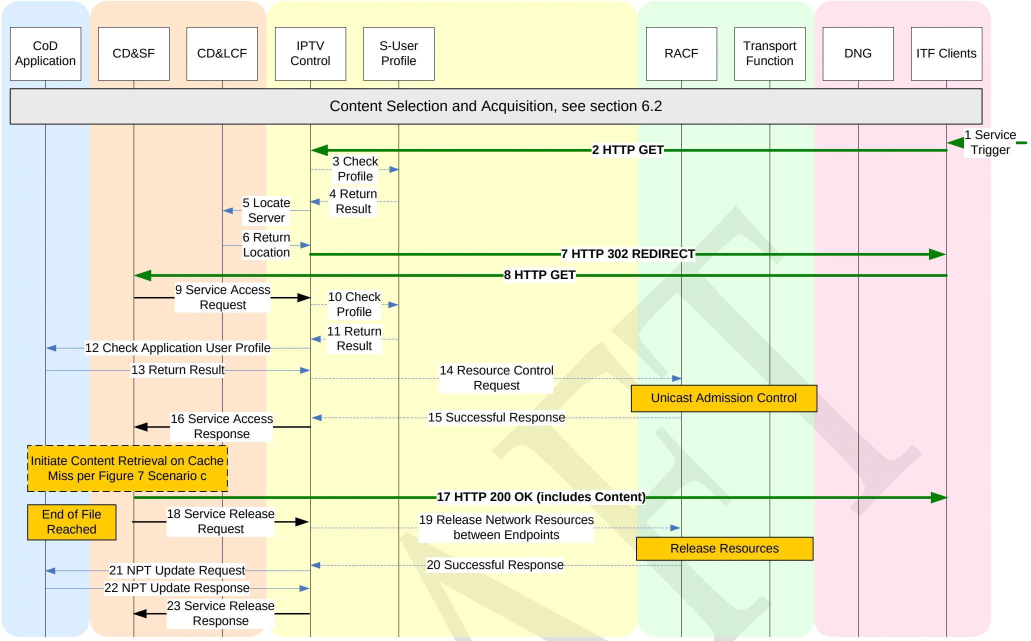


**Figure 19: Non-IMS Redirect CoD Session – HTTP Content Download**

Below is a brief description of the steps in the non-IMS redirect CoD session message flow with HTTP content download shown in Figure 19.

1. The CoD Service is triggered through user interaction, preset recording instructions or other means.
2. The ITF sends an HTTP GET to the IPTV Control Functions, including the OriginContentId provided by the CoD Application Function when the content was selected (E3 setup request per Table 9).
3. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
4. The S-User Profile returns the profile data which the IPTV Control Functions use to validate the service connection (S6) (optional).
5. The IPTV Control Functions request the location of a server within the CD&SF for content delivery from the CD&LCF (S1 locate request per Table 21).
6. The CD&LCF returns the location (CD&SF host) of the chosen server within the CD&SF to the IPTV Control Functions (S1 locate response per Table 21).
7. The IPTV Control Functions return an HTTP 302 REDIRECT to the ITF containing the location (E6 URI) of the chosen server within the CD&SF (E3 setup response per Table 9).
8. The ITF sends an HTTP GET to the specified server location in the CD&SF (E6 download request per Table 20).
9. Since the HTTP GET is outside the context of an E6 session, the CD&SF creates an S5 session for the duration of the HTTP GET transaction. The CD&SF informs the IPTV Control Functions that a request for CoD Service has been made by the ITF (S5 access request per Table 25).

10. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
11. The S-User Profile returns the profile data which the IPTV Control Functions use to validate the service connection (S6) (optional).
12. The IPTV Control Functions request the subscriber's authorization for delivery of the asset (A8 authorization request per Table 3).
13. If the subscriber is authorized, the Application Function returns the allocated session bandwidth and, optionally, content protection information (e.g., ECM) and NPT value (A8 authorization response per Table 3).
14. The IPTV Control Functions *may* initiate a resource reservation request to the RACF based on the bandwidth for the requested CoD Service stream (S3) (optional). The RACF allocates the required bandwidth for the service. The mechanism to identify the value of the bandwidth to be requested is for further study.
15. The RACF signals that the bandwidth allocation is complete to the IPTV Control Functions (S3) (optional).
16. The IPTV Control Functions signal the approval of the CoD Service to the CD&SF (S5 access response per Table 25). Content protection information (e.g., ECM) may be returned and if the CD&SF is expected to enforce a bit rate limit on the download, a maximum bit rate is returned.
17. The CD&SF returns HTTP 200 OK to the ITF, including the content file in the message (E6 download response per Table 20).
18. Once the entire file has been transmitted and the end of file has been reached, the CD&SF requests that the CoD Service session be released by the IPTV Control Functions referencing the SessionId returned in step 16 (S5 release request per Table 25).
19. The IPTV Control Functions request the RACF to deallocate the network resources for the CoD Service in the transport network (S3) (optional). The RACF deallocates the bandwidth needed by the transport function.
20. The RACF signals that the bandwidth deallocation is complete to the IPTV Control Functions (S3) (optional).
21. The IPTV Control Functions update the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
22. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
23. The IPTV Control Functions inform the CD&SF that the session has been released (S5 release response per Table 25).

### 6.13 Redirect CoD Session – HTTP Content Fragmented Transfer

Figure 20, HTTP Content Fragmented Transfer, illustrates explicitly signaled session management using the HTTP POST and DELETE methods. In the illustrated use case, the content is delivered in fragments. Each fragment is retrieved by the ITF client from the CD&SF using a separate HTTP GET transaction. The content Origin URI references the manifest file of the fragmented asset. The manifest file contains the list of content fragment URIs in time sequence corresponding to each fragment. The ITF client retrieves the fragments as needed to deliver a smooth playout of the selected content to the user. This case also can be used to provide adaptive streaming rates. Here each content fragment is encoded with multiple rates with the ITF client selecting the rate associated with a fragment based on the amount of bandwidth allocated for the session.

Network resources are allocated at the beginning of the session via the HTTP POST method. The HTTP Cookie that is returned in the response to the POST contains a SessionId. The ITF uses this Cookie in all subsequent HTTP GET transactions to reference the session that was created via the HTTP POST. Resources for this session are deallocated using the HTTP session termination flow illustrated in section 6.14.

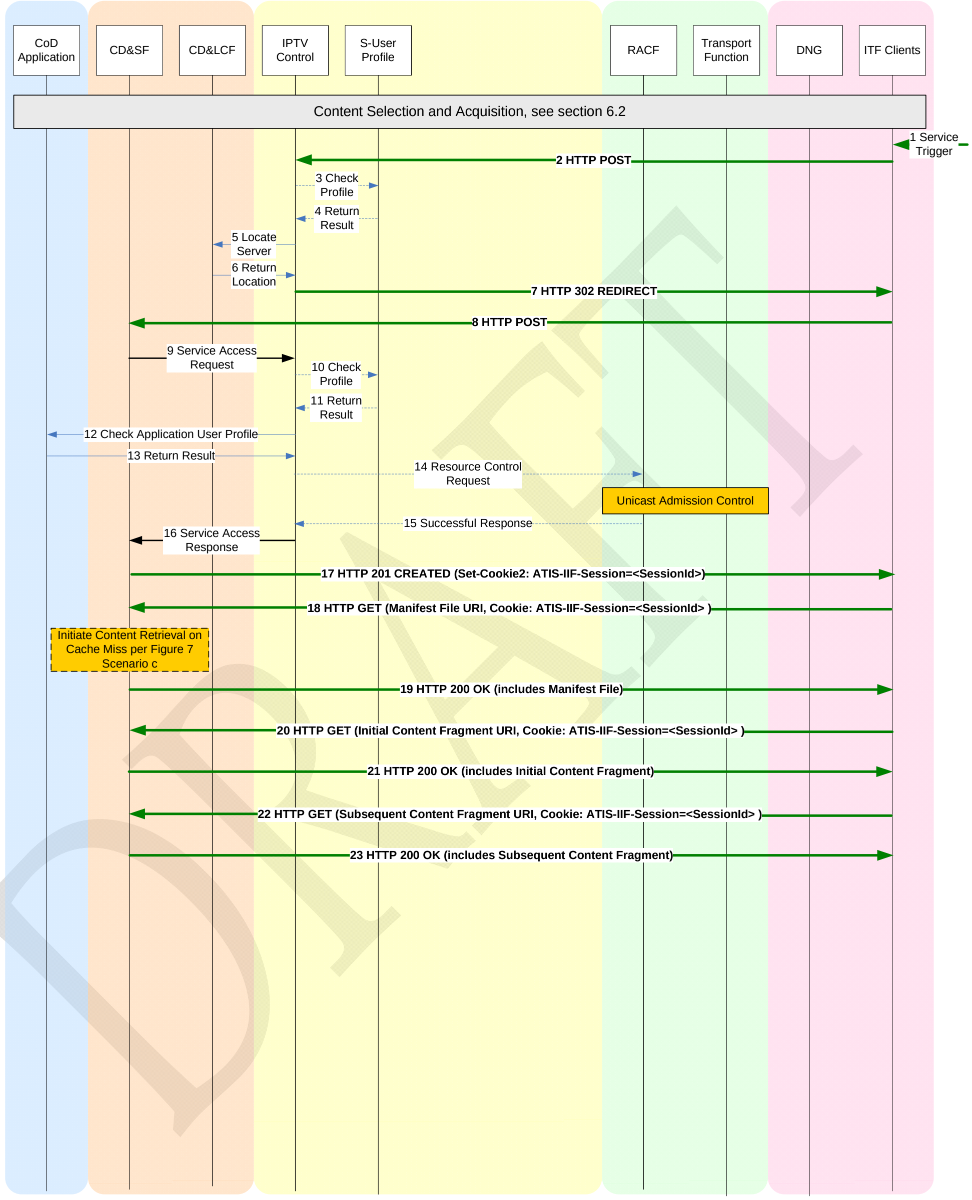

CoD Application
CD&SF
CD&LCF
IPTV Control
S-User Profile
RACF
Transport Function
DNG
ITF Clients
Content Selection and Acquisition, see section 6.2
1 Service Trigger
2 HTTP POST
3 Check Profile
4 Return Result
5 Locate Server
6 Return Location
7 HTTP 302 REDIRECT
8 HTTP POST
9 Service Access Request
10 Check Profile
11 Return Result
12 Check Application User Profile
13 Return Result
14 Resource Control Request
Unicast Admission Control
15 Successful Response
16 Service Access Response
17 HTTP 201 CREATED (Set-Cookie2: ATIS-IIF-Session=<SessionId>)
18 HTTP GET (Manifest File URI, Cookie: ATIS-IIF-Session=<SessionId> )
Initiate Content Retrieval on Cache Miss per Figure 7 Scenario c
19 HTTP 200 OK (includes Manifest File)
20 HTTP GET (Initial Content Fragment URI, Cookie: ATIS-IIF-Session=<SessionId> )
21 HTTP 200 OK (includes Initial Content Fragment)
22 HTTP GET (Subsequent Content Fragment URI, Cookie: ATIS-IIF-Session=<SessionId> )
23 HTTP 200 OK (includes Subsequent Content Fragment)

**Figure 20: Non-IMS Redirect CoD Session – HTTP Content Fragmented Transfer**

Below is a brief description of the steps in the non-IMS redirect CoD session message flow with HTTP content fragmented transfer shown in Figure 20

1. The CoD Service is triggered through user interaction, preset recording instructions or other means.
2. The ITF sends an HTTP POST to the IPTV Control Functions, including the OriginContentId provided by the CoD Application Function when the content was selected (E3 setup request per Table 9).
3. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
4. The S-User Profile returns the profile data which the IPTV Control Functions use to validate the service connection (S6) (optional).
5. The IPTV Control Functions request the location of a server within the CD&SF for content delivery from the CD&LCF (S1 locate request per Table 21).
6. The CD&LCF returns the location of the chosen server (CD&SF host) within the CD&SF to the IPTV Control Functions (S1 locate response per Table 21).
7. The IPTV Control Functions return an HTTP 302 REDIRECT to the ITF containing the location of the chosen server (E6 URI) within the CD&SF (E3 setup response per Table 9). Since no session state was allocated, this response does not include a Cookie with the SessionId.
8. The ITF sends an HTTP POST to the URI specified in the Location header from step 7 (E6 setup request per Table 20).
9. The CD&SF informs the IPTV Control Functions that a request for CoD Service has been made by the ITF (S5 access request per Table 25).
10. The IPTV Control Functions request the user's S-User Profile (S6) (optional).
11. The S-User Profile returns the profile data which the IPTV Control Functions uses to validate the service connection (S6) (optional).
12. The IPTV Control Functions request the subscriber's authorization for delivery of the asset (A8 authorization request per Table 3).
13. If the subscriber is authorized, the Application Function returns the allocated session bandwidth and, optionally, content protection information (e.g., ECM) and NPT value (A8 authorization response per Table 3).
14. The IPTV Control Functions initiate a resource reservation request to the RACF based on the bandwidth for the requested CoD Service stream (S3) (optional). The RACF allocates the required bandwidth for the service.
15. The RACF signals that the bandwidth allocation is complete to the IPTV Control Functions (S3) (optional).
16. The IPTV Control Functions signal the approval of the CoD Service to the CD&SF (S5 access response per Table 25). Content protection information (e.g., ECM) may be communicated and if the CD&SF is expected to enforce a bit rate limit on the download, a maximum bit rate is communicated.
17. The CD&SF returns HTTP 201 Created to the ITF (E6 setup response per Table 20). Since session creation was successful, the response includes a Set-Cookie2 header with the assigned SessionId, the CD&SF Session URI that the ITF uses to tear down the session, and the bandwidth allocated for the session. The ITF may use the returned bandwidth value to select a set of fragments corresponding to the allocated rate.
18. The ITF issues an HTTP GET using the URI returned in the Location header from step 7 and the Set-Cookie2 returned in step 17 (E6 download request per Table 20).
19. The CD&SF returns HTTP 200 OK, including the manifest file in the message (E6 download response per Table 20).
20. The ITF sends an HTTP GET along with the session Cookie to retrieve the initial content fragment, using the location from the manifest file for the initial fragment (E6 download request per Table 20).

21. The CD&SF returns HTTP 200 OK to the ITF, (E6 download response per Table 20).
22. For each remaining fragment of the content, the ITF issues an HTTP GET to the CD&SF using the location in the manifest file for the fragment along with the session Cookie (E6 download request per Table 20).
23. The CD&SF returns HTTP 200 OK, containing the content fragment (E6 download response per Table 20

### 6.14 Redirect CoD Session – HTTP Session Termination

Figure 21, HTTP session termination illustrates the flow used to release resources associated with the session management use case illustrated in section 6.13. The ITF signals the end of the session with an HTTP DELETE method. Session-based resources are released and the HTTP Cookie used to identify the session is discarded by the ITF.

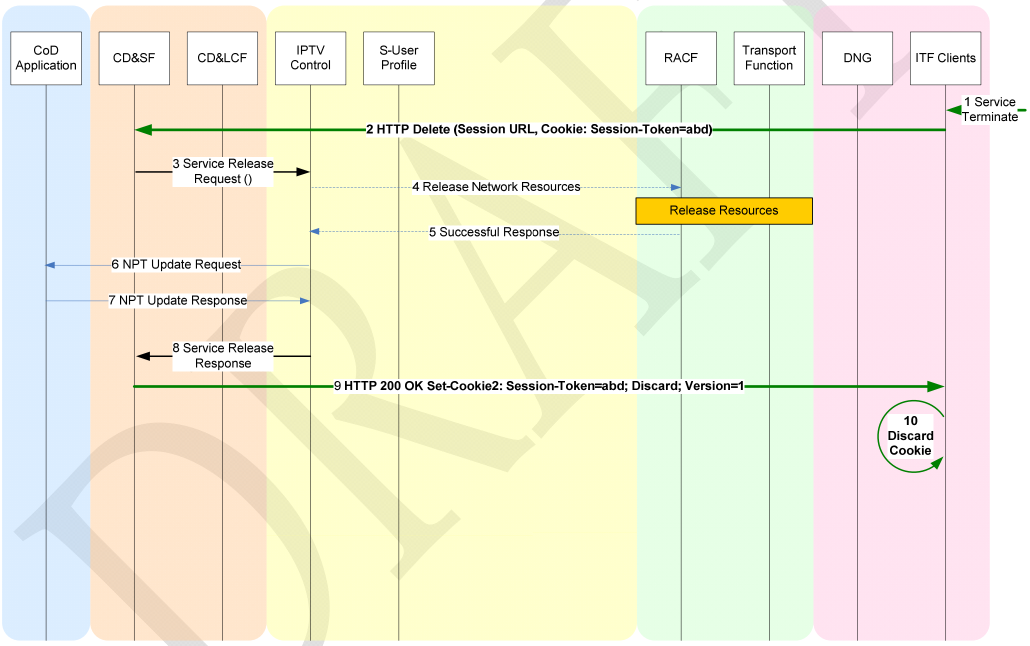


**Figure 21: Redirect CoD Session – HTTP Session Termination**

Below is a brief description of the steps in the HTTP session termination message flow shown in Figure 21.

1. The CoD Service termination is triggered through user interaction or other means.
2. The ITF sends an HTTP DELETE to the CD&SF using the CD&SF Session URI (E6 teardown request per Table 20).
3. The CD&SF tears down the HTTP session and requests that the CoD Service session be released by the IPTV Control Functions (S5 release request per Table 25).

4. The IPTV Control Functions request the RACF to deallocate network resources for the CoD service in the transport network (S3) (optional). The RACF deallocates the bandwidth needed by the transport function.
5. The RACF signals that the bandwidth deallocation is complete to the IPTV Control Functions (S3) (optional).
6. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
7. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
8. The IPTV Control Functions inform the CD&SF that the session has been released (S5 release response per Table 25).
9. The CD&SF sends HTTP 200 OK to the ITF (E6 teardown response per Table 20). The discard parameter of the Set-Cookie2 header in the response is an indication to the ITF to discard the Cookie that was used to identify the session.
10. The ITF discards the Cookie. The service is now terminated.

### *6.15 IMS CoD Session Establishment*

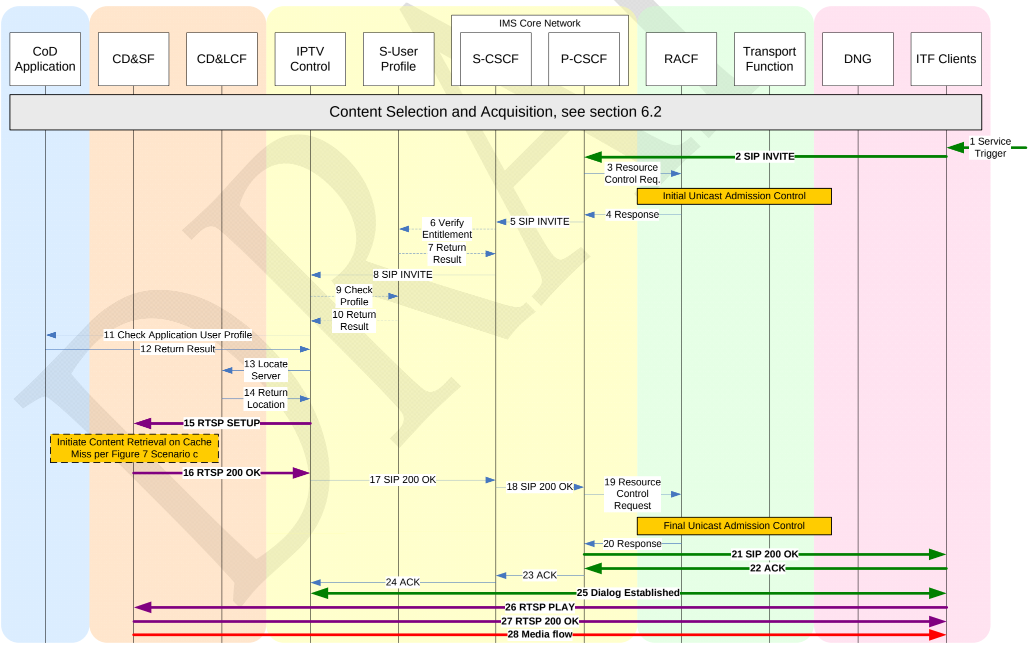


**Figure 22: IMS CoD Session Establishment**

Below is a brief description of the steps in the IMS CoD session establishment message flow shown in Figure 22.

1. The CoD Service is triggered through user interaction, preset recording instructions or other means.
2. The ITF initiates a SIP session by sending a SIP INVITE to the P-CSCF (E3). The INVITE includes the OriginContentId provided by the CoD Application Function when the content was selected.
3. The P-CSCF initiates a resource reservation request (reservation phase) to the RACF based on the bandwidth requested in the incoming INVITE. The RACF allocates the default initial bandwidth (S3).
4. The RACF signals that the bandwidth allocation is complete to the P-CSCF (S3).
5. The P-CSCF forwards the INVITE to the S-CSCF.
6. S-CSCF verifies the user's entitlement from the S-User Profile (S2) (optional).
7. The S-User Profile provides subscription information and determines the IPTV Control Functions to which the user has subscribed. If the subscriber is not entitled to IPTV services, the S-CSCF rejects the SIP session request (optional).
8. The S-CSCF forwards the INVITE to the IPTV Control Functions (S7).
9. The IPTV Control Functions request the user's S-User Profile if not available (S6) (optional).
10. The S-User Profile returns the profile data which the IPTV Control Functions use to validate the service connection (S6) (optional).
11. The IPTV Control Functions request the subscriber's authorization for delivery of the asset (A8 authorization request per Table 3).
12. If the subscriber is authorized, the Application Function returns the allocated session bandwidth and, optionally, content protection information (e.g., ECM) and NPT value (A8 authorization response per Table 3).
13. The IPTV Control Functions request the location of a server within the CD&SF for content delivery from the CD&LCF (S1 locate request per Table 21).
14. The CD&LCF returns the location of the chosen server (CD&SF host) within the CD&SF to the IPTV Control Functions (S1 locate response per Table 21).
15. The IPTV Control Functions send an RTSP SETUP to the specified server location in the CD&SF (S5 setup request per Table 22).
16. The CD&SF establishes the RTSP session and returns an RTSP 200 OK to the IPTV Control Functions including the RTSP MediaSessionId (S5 setup response per Table 22).
17. The IPTV Control Functions return SIP 200 OK to the S-CSCF to acknowledge the INVITE, including the RTSP MediaSessionId and the location of the server within the CD&SF (S7).
18. The S-CSCF returns SIP 200 OK to the P-CSCF, including the RTSP MediaSessionId and the location of the server within the CD&SF.
19. The P-CSCF initiates a resource reservation request (commitment phase) to the RACF based on the bandwidth for the requested CoD Service stream (S3). The RACF allocates the required bandwidth for the service.
20. The RACF signals that the bandwidth allocation is complete to the P-CSCF (S3).
21. The P-CSCF returns SIP 200 OK to the ITF, including the RTSP MediaSessionId and the location of the server within the CD&SF (E3).
22. The ITF sends an ACK to the P-CSCF (E3).
23. The P-CSCF sends an ACK to the S-CSCF.
24. The S-CSCF sends and ACK to the IPTV Control Functions.
25. The SIP dialog is now established between the ITF and the IPTV Control Functions.
26. The ITF sends an RTSP PLAY for the signaled RTSP MediaSessionId to the location of the server within the CD&SF (E6 play request per Table 15).

27. The CD&SF starts media playback within the RTSP session and returns RTSP 200 OK to the ITF (E6 play response per Table 15).
28. The media is now flowing to the ITF (Ud).

### 6.16 IMS CoD Session Termination

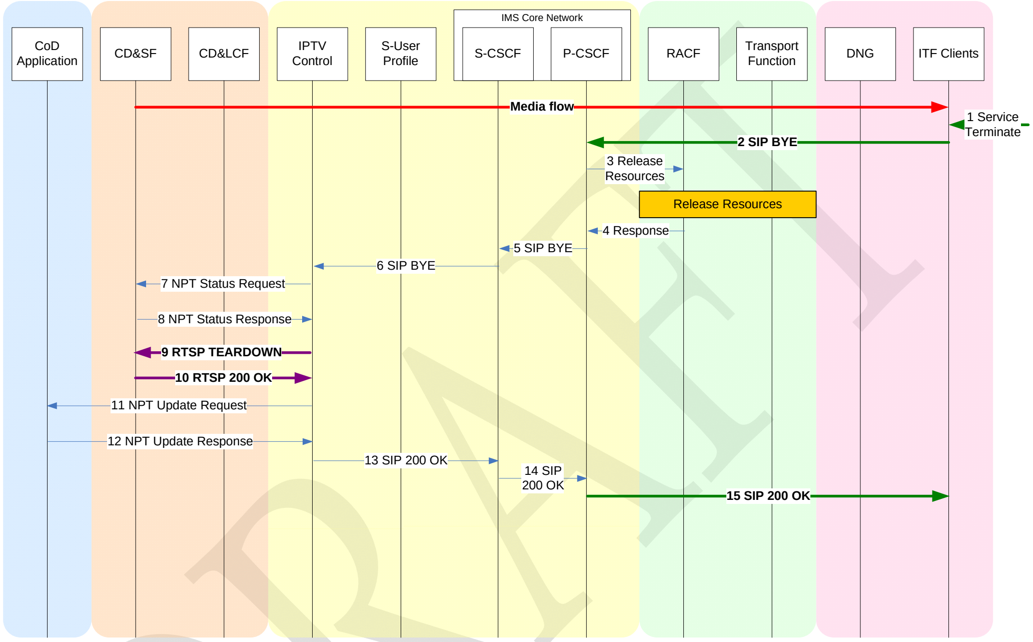


Figure 23: IMS CoD Session Termination

Below is a brief description of the steps in the IMS CoD session termination message flow shown in Figure 23.

1. The CoD service termination is triggered through user interaction or other means.
2. The ITF sends a SIP BYE to the P-CSCF (E3).
3. The P-CSCF requests the RACF to deallocate the network resources for the CoD Service in the transport network (S3). The RACF deallocates the bandwidth needed by the transport function.
4. The RACF signals that the bandwidth deallocation is complete to the P-CSCF (S3).
5. The P-CSCF forwards the SIP BYE to the S-CSCF.
6. The S-CSCF forwards the SIP BYE to the IPTV Control Functions.
7. The IPTV Control Functions request the NPT value from the CD&SF (S5 status request per Table 22).
8. The CD&SF returns the NPT value (S5 status response per Table 22).
9. The IPTV Control Functions send an RTSP TEARDOWN to the CD&SF (S5 teardown request per Table 22).

10. The CD&SF tears down the RTSP session and returns RTSP 200 OK to the IPTV Control Functions (S5 teardown response per Table 22).
11. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
12. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).
13. The IPTV Control Functions send SIP 200 OK to the S-CSCF.
14. The S-CSCF sends SIP 200 OK to the P-CSCF.
15. The P-CSCF sends SIP 200 OK to the ITF (E3). The dialog is now closed and the SIP session is terminated.

### 6.17 IMS CD&SF Initiated Session Termination

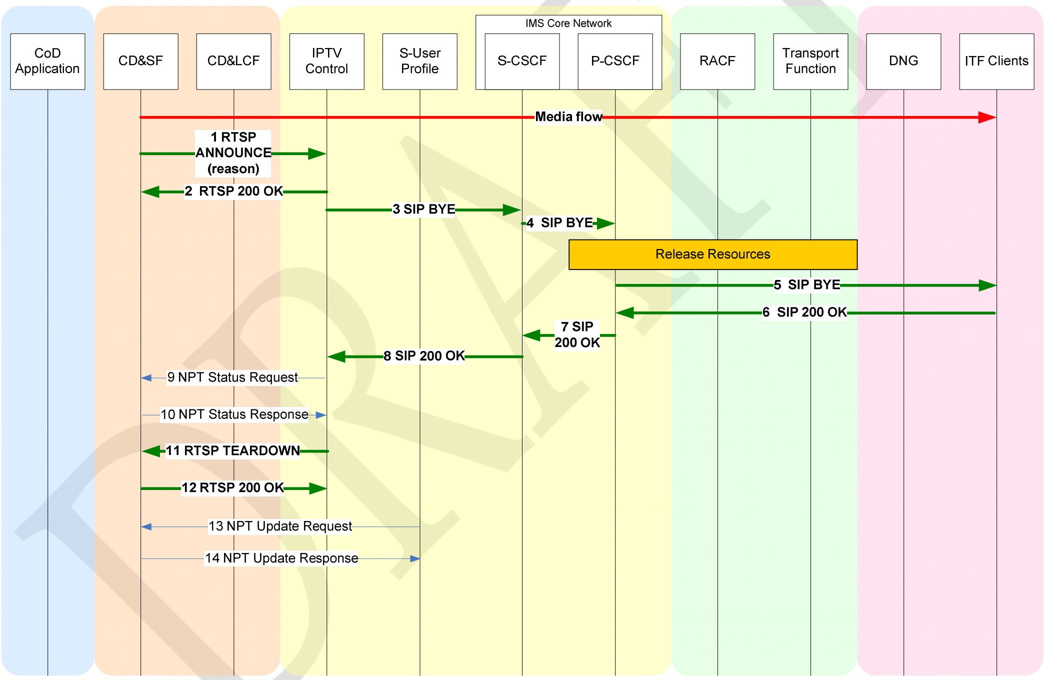


**Figure 24: IMS CD&SF Initiated Session Termination**

Below is a brief description of the steps in the IMS CD&SF initiated session termination message flow shown in Figure 24. As a pre-requisite it is assumed that the media is flowing to the ITF and there is an RTSP session in place between the IPTV Control Functions and the CD&SF.

1. The CD&SF disconnects locally the RTSP session and sends an RTSP ANNOUNCE to the IPTV Control Functions to announce that the RTSP has been terminated and the reason for termination (S5 event request per Table 22).

2. The IPTV Control Functions acknowledge the RTSP ANNOUNCE and returns an RTSP 200 OK to the CD&SF (S5 event response per Table 22).
3. The IPTV Control Functions locate the SIP session associated with the RTSP session and sends a SIP BYE to the S-CSCF.
4. The S-CSCF proxies the SIP BYE to the P-CSCF.  The P-CSCF releases the network resources associated with the SIP session.
5. Following the successfully release of the network resources, the P-CSCF sends a SIP BYE to the ITF client (E3).
6. The ITF clients responds with a SIP 200 OK acknowledging the SIP BYE (E3).
7. The P-CSCF proxies the SIP 200 OK to the S-CSCF.
8. The S-CSCF proxies the SIP 200 OK to the IPTV Control Functions.
9. The IPTV Control Functions request the NPT value from the CD&SF (S5 status request per Table 22).
10. The CD&SF returns the NPT value (S5 status response per Table 22).
11. The IPTV Control Functions optionally sends an RTSP TEARDOWN to the CD&SF (S5 teardown request per Table 22).
12. The CD&SF acknowledges the RTSP TEARDOWN and returns an RTSP 200 OK to the IPTV Control Functions (S5 teardown response per Table 22).  The service is now terminated.
13. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
14. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).

## 6.18 IMS IPTV Control Functions Initiated Session Termination

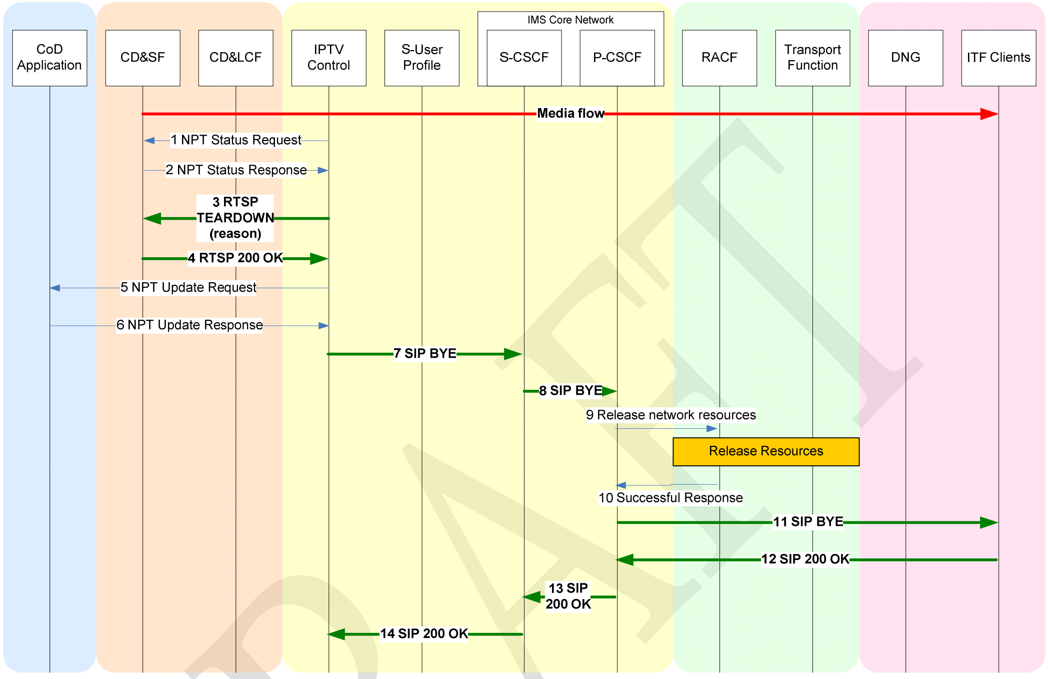


**Figure 25: IMS IPTV Control Functions Initiated Session Termination**

Below is a brief description of the steps in the IMS CoD Service Control initiated session termination message flow shown in Figure 25. As a pre-requisite it is assumed that the media is flowing to the ITF and there is an RTSP session in place between the IPTV Control Functions and the CD&SF.

1. The IPTV Control Functions request the NPT value from the CD&SF (S5 status request per Table 22).
2. The CD&SF returns the NPT value (S5 status response per Table 22).
3. The IPTV Control Functions disconnects locally the RTSP session and sends an RTSP TEARDOWN to the CD&SF to announce that the RTSP has been terminated and the reason for termination (S5 teardown request per Table 22).
4. The CD&SF acknowledges the RTSP TEARDOWN and returns an RTSP 200 OK to the IPTV Control Functions (S5 teardown response per Table 22).
5. The IPTV Control Functions provide an update of the NPT playout position of the content to the CoD Application Function (A8 NPT update request per Table 3).
6. The CoD Application Function acknowledges the NPT update (A8 NPT update response per Table 3).

7. The IPTV Control Functions locates the SIP session associated with the RTSP session and sends a SIP BYE to the S-CSCF.
8. The S-CSCF proxies the SIP BYE to the P-CSCF. The P-CSCF releases the network resources associated with the SIP session.
9. The P-CSCF requests the RACF to release the network resources associated with the session (S3).
10. The RACF acknowledges release of the resources (S3).
11. Following the successfully release of the network resources, the P-CSCF sends a SIP BYE to the ITF client (E3).
12. The ITF clients responds with a SIP 200 OK acknowledging the SIP BYE (E3).
13. The P-CSCF proxies the SIP 200 OK to the S-CSCF.
14. The S-CSCF proxies the SIP 200 OK to the IPTV Control Functions.

# 7 URI FORMATS AND USAGE

## 7.1 Overview

Figures 10 and 22 illustrate proxy session setup flows for RTSP and IMS respectively, while Figures 15 and 19 illustrate the RTSP redirect and HTTP session flows, respectively. In each of these flows, the CD&SF may need to perform a cache fill operation back to the Content Origin Function as the result of an RTSP or HTTP session request arriving via reference point S5 (RTSP proxy or IMS) or E6 (RTSP redirect or HTTP). To perform the cache fill operation, the CD&SF must be able to map the RTSP or HTTP request URI provided via reference point S5 or E6 to an HTTP request URI that is used on reference point C2 to the Content Origin Function.

This section describes the URIs used on reference points to convey signaling information from one CoD functional component to another. To better understand the translation of URIs from one reference point to another, this section is organized to follow the basic message flows of section 0. Using ABNF for the URI syntax specification, one can follow the URI translation from one reference point to the next. An example URI flow using these translations is included at the end of this section.

The OriginContentId, consisting of OriginId/ContentId as specified in section 3.2.5, is central to the construction and translation of reference point URIs. An OriginContentId is assigned by the Asset Preparation Function for each asset. The OriginId component *shall* correspond to the HTTP host name of the Content Origin Function for that asset and the ContentId *shall* correspond to the HTTP abs_path for the asset on the Content Origin Function host. In this way, any functional element with access to the OriginContentId can easily determine the origin for a asset.

Example OriginContentIds include:

| OriginId | ContentId | OriginContentId |
|---|---|---|
| origin.sp.com | movies/hd/big.m2ts | origin.sp.com/movies/hd/big.m2ts |
| origin.sp.com | cp.com/movies/hd/big.m2ts | origin.sp.com/cp.com/movies/hd/big.m2ts |

## 7.2 Origin URI

The Origin URI is used by the C2 reference point to reference resources on the Content Origin Function. The CD&SF uses the C2 reference point to retrieve resources from the Content Origin

Function. The defined syntax of the Origin URI enables the CD&SF to reference resources stored anywhere in the World Wide Web.

The syntax of the Origin URI is specified as follows:

```
OriginURI  = "http:" / "https:" hier-part
hier-part  = "//" host "/" path-absolute
host = reg-name                    ; reg-name specified in section 3.2.2 of RFC 3986
path-absolute is specified in section 3.3 of RFC 3986.
```

Example Origin URIs:

http://origin.sp.com/movies/hd/big.m2ts

http://origin.sp.com/cp.com/movies/hd/big.m2ts

The OriginContentId, used to refer to an asset, can be translated into an Origin URI by prepending the http scheme to the OriginContentId. The CD&SF uses this translation to translate OriginContentIds provided via the S5 and E6 reference points into the Origin URI used by the C2 reference point.

Example OriginContentIds and their associated Origin URI include:

| OriginId | ContentId | Origin URI |
|---|---|---|
| origin.sp.com | movies/hd/big.m2ts | http://origin.sp.com/movies/hd/big.m2ts |
| origin.sp.com | cp.com/movies/hd/big.m2ts | http://origin.sp.com/cp.com/movies/hd/big.m2ts |

### *7.3 C5 URI*

The Asset Preparation Functions use the C5 reference point to create and delete resources on the Content Origin Function. The usage of query strings in the C5 reference point is specified in section 10.6.

The syntax of the C5 URI is specified as follows:

```
C5URI = C5scheme "://" COF-host "/" "ATIS-IIF-Assets" ("/" OriginContentId / "?" C5query )

C5scheme = "http" / "https"
C5query = ”max=” 1*<DIGIT> [ “&start=” OriginContentId ]
COF-host = reg-name           ; reg-name specified in section 3.2.2 of RFC 3986
```

Example C5 URIs:

http://cof.sp.com/ATIS-IIF-Assets/origin.sp.com/movies/hd/big.m2ts
;for create, delete and get functions

http://cof.sp.com/ATIS-IIF-Assets/?max=100&start=origin.sp.com/movies/hd/big.m2ts
;for list function

### 7.4 A3 URI

The A3 URI consists of an HTTP URI syntax conveyed to the Asset Preparation Functions by the CoD Application Function through the A3 reference point that enables asset metadata to flow from the Asset Preparation Functions to the CoD Application Function.

Note that the data passed over A3 consists of assets that consist of metadata that may contain references to other assets, including assets that are on the Content Origin Function. Included in the metadata is the OriginContentId for the asset. The Application Function does not generally request assets.

The syntax of the A3 URI is specified as follows:

```
A3URI  = A3scheme "://" APF-host "/" "ATIS-IIF-Assets" ["/" CreatorAssetId ] ["?" A3query ]
A3scheme = "http" / "https"
A3query = query                  ; query specified in section 3.4 of RFC 3986
APF-host = reg-name              ; reg-name specified in section 3.2.2 of RFC 3986
```

Example A3 URIs:

```
http://apf.sp.com/ATIS-IIF-Assets                                      ; for list function
http://apf.sp.com/ATIS-IIF-Assets/cp.com/movies/hd/big.xml             ; for get function
```

### 7.5 C1 URI

The C1 URI consists of an HTTP URI syntax conveyed to the Distribution Control Function of the CD&LCF by the Content Origin Function through the C1 reference point that enables the Content Origin Function to create and delete resources on the CD&SF via D1.

The syntax of the C1 URI is specified as follows:

```
C1URI  = C1scheme "://" DCF-host "/" "ATIS-IIF-Assets" ("/" OriginContentId / "?" C1query )
C1scheme = "http" / "https"
C1query = ”max=” 1*<DIGIT> [ “&start=” OriginContentId ]
DCF-host = reg-name              ; reg-name specified in section 3.2.2 of RFC 3986
```

Example C1 URIs:

```
http://dcf.sp.com/ATIS-IIF-Assets/origin.sp.com/movies/hd/big.m2ts
          ;for create, delete and get functions
http://dcf.sp.com/ATIS-IIF-Assets/?max=100&start=origin.sp.com/movies/hd/big.m2ts
          ;for list function
```

### 7.6 E1 URI

The E1 URI consists of a single URI syntax that combines the RTSP, HTTP and SIP URI syntaxes. The E1 URI is conveyed to the ITF by the CoD Application Function through the E1 reference point. The ITF uses E1 URIs (e.g., when communicating with the IPTV Service Control Function) to reference Content Origin Function resources.

The E1 URI enables an HTTP resource located at the Content Origin Function to be referenced via RTSP, HTTP and SIP. A common URI format is used across RTSP, HTTP and SIP for referencing resources on the Content Origin Function. The syntax of the E1 URI is specified as follows:

```
E1URI  = subscriber-scheme "://" IPTV-SCF-host "/" OriginContentId ["?"E1query]
subscriber-scheme = "rtsp" / "http" / "https" / "sip"
E1query = query                 ; query specified in section 3.4 of RFC 3986
IPTV-SCF-host = reg-name        ; reg-name specified in section 3.2.2 of RFC 3986
```

The path component of an E1 URI is a specific instance of a path as defined in section 3.3 of RFC 3986, and is composed of the OriginContentId.

Example E1 URIs:

rtsp://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts

http://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts

### 7.7 E3 URI

The E3 URI consists of a single URI syntax that combines the RTSP, HTTP and SIP URI syntaxes. An ITF establishes a session using the E3 URI through the following process. The ITF uses the scheme of the E1 URI to determine the session protocol (RTSP, HTTP or SIP) that must be used to access resources via the E3 reference point. The ITF uses the E1 URI as the base of the request URI in the specified session protocol. The E3 URI consists of the E1 URI with an optional query string that is concatenated to the E1 URI. The E3 URI is used to reference Content Origin Function resources and to convey ITF-related information via the E3 reference point.

The E3 URI enables an HTTP resource located at the Content Origin Function to be referenced via an RTSP, HTTP and SIP URI. A common URI format is used across RTSP, HTTP and SIP for referencing resources on the Content Origin Function. The syntax of the E3 URI is specified as follows:

```
E3URI  = subscriber-scheme "://" IPTV-SCF-host "/" OriginContentId "?" E3query
subscriber-scheme = "rtsp" / "http" / "https" / "sip"
E3query = [E1query "&"] / query
IPTV-SCF-host = reg-name        ; reg-name specified in section 3.2.2 of RFC 3986
```

For the SIP URI syntax, refer to sections 10.8.3.3 and 10.8.3.4 for the construction of the E3 URI by the ITF.

Example E3 URIs:

rtsp://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts

http://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts

### 7.8 A8 URI

The A8 URI consists of an HTTP URI syntax conveyed to the CoD Application Function by the IPTV Control Functions via the A8 reference point. The A8 URI is constructed using the OriginContentId, and either the E3 query in the case of RTSP proxy or the S5 query in the case of RTSP redirect or HTTP delivery. The syntax of the A8 URI is specified as follows:

```
A8URI = A8scheme "://" COD-AF-host "/ OriginContentId “?” E3query / S5query
A8scheme = "http" / "https"
COD-AF-host = reg-name      ; reg-name specified in section 3.2.2 of RFC 3986
```

Example A8 URI:

```
http://cod-af.sp.com/origin.sp.com/movies/hd/big.m2ts?ATIS-IIF-
MinBandwidth=3750000&ATIS-IIF-MaxBandwidth=4000000
```

### *7.9 S1 URI*

The S1 URI consists of an HTTP URI syntax conveyed to the Location Control Function of the CD&LCF by the IPTV Control Functions via the S1 reference point. The S1 URI is constructed from the E3 URI by replacing the host portion of the E3 URI with the CD&LCF host and including the ITF-IP-Address query. The syntax of the S1 URI is specified as follows:

```
S1URI  = S1scheme "://" CD&LCF-host "/" OriginContentId "?" S1query
S1scheme = "http" / "https"
S1query = "ATIS-IIF-Subscriber-Scheme=" subscriber-scheme "&ATIS-IIF-ITF-IP-Address="
IPv4address / IPv6address
        ; subscriber-scheme = "rtsp" / "http" / "https" / "sip" as indicated in E3 URI
        ;IPv4address and IPv6address specified in section 3.2.2 of RFC 3986
CD&LCF-host = reg-name      ; reg-name specified in section 3.2.2 of RFC 3986
```

Example S1 URI:

```
http://cdlcf.sp.com/origin.sp.com/movies/hd/big.m2ts?ATIS-IIF-Subscriber-Scheme=rtsp
&ATIS-IIF-ITF-IP-Address=10.1.2.3
```

### *7.10 S5 Proxy URI*

The S5 Proxy URI consists of an RTSP URI syntax conveyed to the CD&SF by the CoD Service Control Function through the S5 reference point in the RTSP proxy and IMS cases. For the RTSP proxy case, the S5 Proxy URI is constructed from the RTSP E3 URI by replacing the host portion of the E3 URI with the CS&SF host and adding the S5 Proxy query. The syntax of the S5 Proxy URI is specified as follows:

```
S5ProxyURI  = S5Proxy-scheme "://" CD&SF-host "/" OriginContentId "?" S5Proxy-query
S5Proxy-scheme = "rtsp"
S5Proxy-query = E3query
CD&SF-host = reg-name       ; reg-name specified in section 3.2.2 of RFC 3986
```

Example S5 Proxy URI:

```
rtsp://cdsf.sp.com/origin.sp.com/movies/hd/big.m2ts
```

### 7.11 E6 URI

The E6 URI consists of a single URI syntax that combines the RTSP and HTTP URI syntaxes. The E6 URI is created by the IPTV Control Functions from the E3 URI, with the host component replaced by the CD&SF host. The CD&SF URI is conveyed to the ITF in the RTSP or HTTP Location header of the E3 reference point response. The syntax of the E6 URI is specified as follows:

```
E6URI  = E6scheme "://" CD&SF-host "/" OriginContentId "?" E6query
E6scheme = "http" / "https" / "rtsp"
E6query = E3query ["&" query]
CD&SF-host = reg-name        ; reg-name specified in section 3.2.2 of RFC 3986
```

Example E6 URIs:

```
rtsp://cdsf.sp.com/origin.sp.com/movies/hd/big.m2ts
http://cdsf.sp.com/origin.sp.com/movies/hd/big.m2ts
```

### 7.12 S5 Redirect URI

The S5 Redirect URI consists of an HTTP URI syntax conveyed to the CoD Service Control Function by the CD&SF through the S5 reference point in the RTSP redirect and HTTP delivery cases. The S5 Redirect URI is constructed from the E6 URI by replacing the host portion of the E6 URI with the CoD Service Control Function host and adding the S5 Redirect query.

The syntax of the S5 Redirect URI for the access function is specified as follows:

```
S5RedirectURI  = S5Redirect-scheme "://" COD-SCF-host "/" OriginContentId "?" S5Redirect-
query
S5Redirect-scheme = "http" / "https"
S5Redirect-query = E6query "&" 7*<reqElement "&"> [query]
reqElement = "ATIS-IIF-MinBandwidth=" MinBandwidth / "ATIS-IIF-MaxBandwidth="
MaxBandwidth / "ATIS-IIF-Subscriber-Scheme=" subscriber-scheme / "ATIS-IIF-SourceIP="
CD&SF IP / "ATIS-IIF-SourcePort=" CD&SF port / "ATIS-IIF-DestIP=" ITF IP / "ATIS-IIF-
DestPort=" ITF port
MinBandwidth = 1*<DIGIT>
MaxBandwidth = 1*<DIGIT>
subscriber-scheme = "rtsp" / "http" / "https"  as indicated in E6URI
CD&SF IP = host              ; host specified in section 3.2.2 of RFC 3986
ITF IP = host                ; host specified in section 3.2.2 of RFC 3986
CD&SF port = port            ; port specified in section 3.2.3 of RFC 3986
ITF port = port              ; port specified in section 3.2.3 of RFC 3986
COD-SCF-host = reg-name      ; reg-name specified in section 3.2.2 of RFC 3986
```

Example S5 Redirect URI for the access function:

```
http://cod-scf.sp.com/origin.sp.com/movies/hd/big.m2ts?ATIS-IIF-
MinBandwidth=3750000&ATIS-IIF-MaxBandwidth=4000000&ATIS-IIF-Subscriber-Scheme=rtsp
```

&ATIS-IIF-SourceIP=10.1.2.3&ATIS-IIF-Sourceport=80&ATIS-IIF-DestIP=10.1.3.4&ATIS-IIF-Destport=500

The syntax of the S5 Redirect URI for the release and query functions is specified as follows:

COD-SCFSessionURI = S5Redirect-scheme "://" COD-SCF-host "/" "ATIS-IIF-Sessions" "/" SessionId

S5Redirect-scheme = "http" / "https"

COD-SCF-host = reg-name ; reg-name specified in section 3.2.2 of RFC 3986

Example S5 Redirect URI for the release function:

http://cod-scf.sp.com/ATIS-IIF-Sessions/v374v83i7t43b378

The syntax of the S5 Redirect URI for the terminate, list, and status functions is specified as follows:

S5RedirectURI = S5Redirect-scheme "://" CD&SF-host "/" "ATIS-IIF-Sessions" ["/" SessionId] ["?" S5query]

S5Redirect-scheme = "http" / "https"

S5query = "ATIS-IIF-S5-Notify=terminate"

CD&SF-host = reg-name ; reg-name specified in section 3.2.2 of RFC 3986

Example S5 Redirect URI for the status function:

http://cdsf.sp.com/ATIS-IIF-Sessions/v374v83i7t43b378

## 7.13 URI Flow Example

The URI flow example in Figure 26 illustrates how the OriginContentId, and the resulting Origin URI associated with an asset, flows through both the content preparation and session establishment flows in the CoD Service architecture. The illustrated message flow is for the non-IMS RSTP redirect CoD session establishment corresponding to Figure 15.

**Deleted:** Figure 15

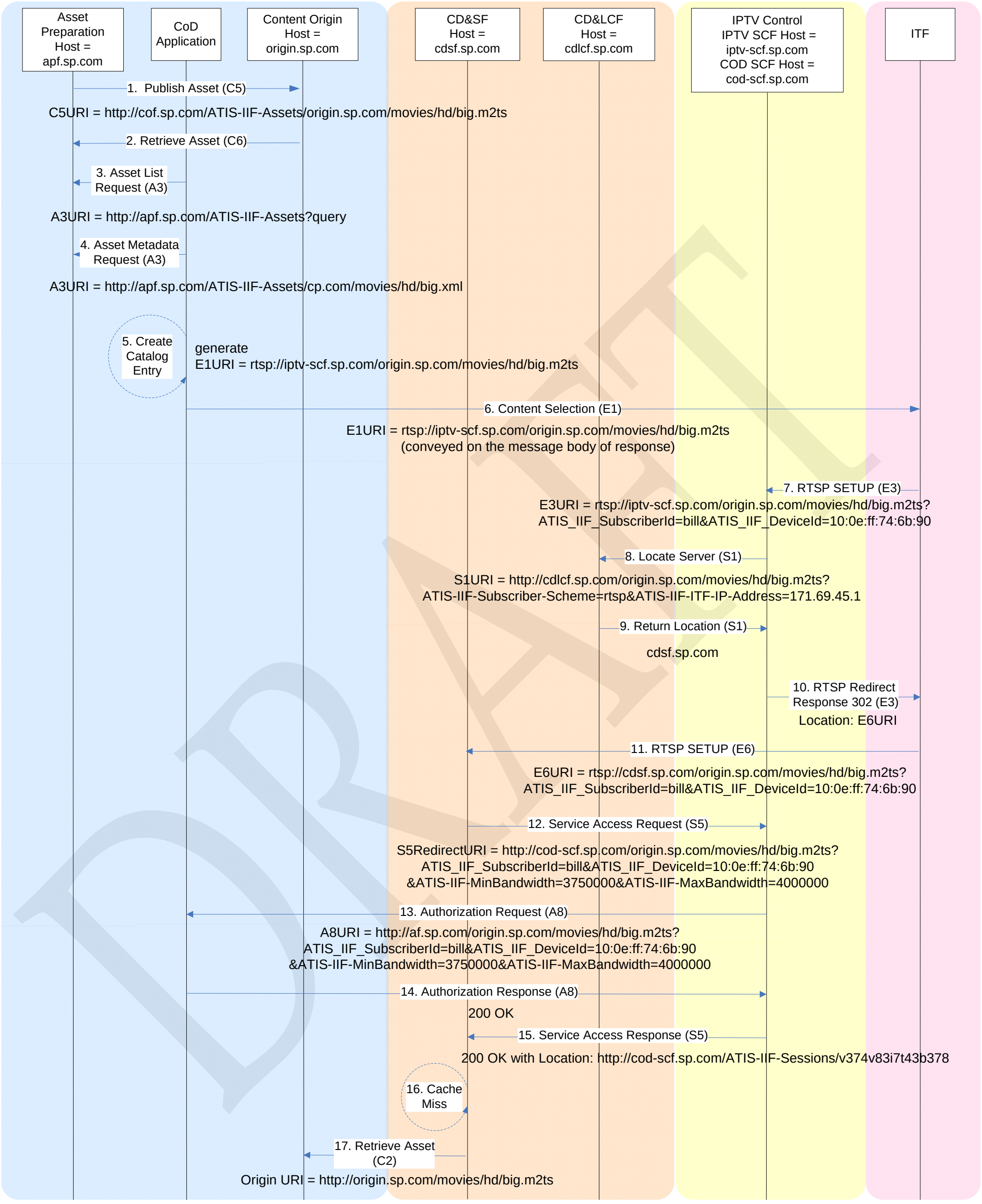

Asset Preparation Host = apf.sp.com
CoD Application
Content Origin Host = origin.sp.com
CD&SF Host = cdsf.sp.com
CD&LCF Host = cdlcf.sp.com
IPTV Control IPTV SCF Host = iptv-scf.sp.com COD SCF Host = cod-scf.sp.com
ITF
1. Publish Asset (C5)
C5URI = http://cof.sp.com/ATIS-IIF-Assets/origin.sp.com/movies/hd/big.m2ts
2. Retrieve Asset (C6)
3. Asset List Request (A3)
A3URI = http://apf.sp.com/ATIS-IIF-Assets?query
4. Asset Metadata Request (A3)
A3URI = http://apf.sp.com/ATIS-IIF-Assets/cp.com/movies/hd/big.xml
5. Create Catalog Entry
generate
E1URI = rtsp://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts
6. Content Selection (E1)
E1URI = rtsp://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts
(conveyed on the message body of response)
7. RTSP SETUP (E3)
E3URI = rtsp://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts?
ATIS_IIF_SubscriberId=bill&ATIS_IIF_DeviceId=10:0e:ff:74:6b:90
8. Locate Server (S1)
S1URI = http://cdlcf.sp.com/origin.sp.com/movies/hd/big.m2ts?
ATIS-IIF-Subscriber-Scheme=rtsp&ATIS-IIF-ITF-IP-Address=171.69.45.1
9. Return Location (S1)
cdsf.sp.com
10. RTSP Redirect Response 302 (E3)
Location: E6URI
11. RTSP SETUP (E6)
E6URI = rtsp://cdsf.sp.com/origin.sp.com/movies/hd/big.m2ts?
ATIS_IIF_SubscriberId=bill&ATIS_IIF_DeviceId=10:0e:ff:74:6b:90
12. Service Access Request (S5)
S5RedirectURI = http://cod-scf.sp.com/origin.sp.com/movies/hd/big.m2ts?
ATIS_IIF_SubscriberId=bill&ATIS_IIF_DeviceId=10:0e:ff:74:6b:90
&ATIS-IIF-MinBandwidth=3750000&ATIS-IIF-MaxBandwidth=4000000
13. Authorization Request (A8)
A8URI = http://af.sp.com/origin.sp.com/movies/hd/big.m2ts?
ATIS_IIF_SubscriberId=bill&ATIS_IIF_DeviceId=10:0e:ff:74:6b:90
&ATIS-IIF-MinBandwidth=3750000&ATIS-IIF-MaxBandwidth=4000000
14. Authorization Response (A8)
200 OK
15. Service Access Response (S5)
200 OK with Location: http://cod-scf.sp.com/ATIS-IIF-Sessions/v374v83i7t43b378
16. Cache Miss
17. Retrieve Asset (C2)
Origin URI = http://origin.sp.com/movies/hd/big.m2ts

**Figure 26: URI Flow for RTSP Redirect Session Establishment**

Below is a brief description of the Figure 26 URI flow for an asset with the following attributes:

**OriginId = origin.sp.com**

**ContentId = movies/hd/big.m2ts**

**OriginContentId = origin.sp.com/movies/hd/big.m2ts**

1. The Asset Preparation Functions create an OriginContentId for the asset and provides the OriginContentId to the Content Origin Function within the path of the URI for the C5 create function. It also provides the Asset Preparation Function URI to be used in the next step.

   **C5URI = http:// cof.sp.com/ATIS-IIF-Assets/origin.sp.com/movies/hd/big.m2ts**

2. The Content Origin pulls the asset from the Asset Preparation Functions via the C6 reference point using the Asset Preparation Function URI and is now prepared to serve it using the Origin URI.
3. The CoD Application Function queries the Asset Preparation Functions for new A3 URIs.
4. The CoD Application Function requests the metadata for the ingested asset from the Asset Preparation Functions using the A3 URI via the A3 reference point.

   **A3URI = http://apf.sp.com/cp.com/movies/hd/big.xml**

5. The CoD Application Function creates an E1 URI from the OriginContentId obtained from the asset metadata and stores it in the catalog. The E1 URI is an rtsp URI constructed from the host name of the IPTV Service Control Function and the OriginContentId of the asset.

   **E1URI = rtsp://iptv-scf.sp.com/origin.sp.com/movies/hd/big.m2ts**

6. The ITF receives the E1 URI for the requested asset from the catalog via the Content Selection and Acquisition Flow.
7. The ITF creates the E3 URI from the E1 URI and sends it to the IPTV Control Functions in the RTSP Setup request.

   **E3URI = rtsp://iptv-scf.sp.com/origin.sp.com/hd/movies/hd/big.m2ts**

8. The IPTV Control Functions create the S1 URI from the E3 URI and sends it to the CD&LCF. The S1 URI is formed by replacing the IPTV Control Functions hostname in the E3 URI with the CD&LCF hostname and adding the ITF IP Address query element. In this example, the ITF IP Address is 171.69.45.1.

   **S1URI = http://cdlcf.sp.com/origin.sp.com/movies/hd/big.m2ts?ATIS-IIF-ITF-IP-Address=171.69.45.1**

9. The CD&LCF selects the best CD&SF to serve the request and returns the selected CD&SF hostname.
10. The IPTV Control Functions return the E6 URI in the Location header of the RTSP redirect response. The E6 URI is created by replacing the IPTV Control Functions hostname in the E3 URI with the CD&SF hostname returned in step 9.

    **E6URI = rtsp://cdsf.sp.com/origin.sp.com/movies/hd/big.m2ts**

11. The ITF re-issues the RTSP redirect with the E6 URI it receives in the Location header of the redirect response.

    **E6URI = rtsp://cdsf.sp.com/origin.sp.com/movies/hd/big.m2ts**

12. The CD&SF sends a service access request to the CoD SCF via reference point S5. The S5 URI is created from the E6 URI by replacing the rtsp scheme with http and replacing the E6 URI hostname

with the CoD SCF hostname and appending the ATIS-IIF-MinBandwidth and ATIS-IIF-MaxBandwidth tags to the query.

**S5URI = http://cod-scf.sp.com/origin.sp.com/movies/hd/big.m2ts? ATIS-IIF-MinBandwidth=3750000&ATIS-IIF-MaxBandwidth=4000000**

13. The CoD SCF sends a subscriber authorization request to the CoD Application Function via reference point A8. The A8 URI is created from the S5 URI by replacing the S5 URI hostname with the CoD Application Function hostname.

    **A8URI = http://cod-af.sp.com/origin.sp.com/movies/hd/big.m2ts? ATIS-IIF-MinBandwidth=3750000&ATIS-IIF-MaxBandwidth=4000000**

14. The CoD Application Function returns a successful response to the subscriber authorization request.
15. The CoD SCF returns a successful response to the service access request.
16. The CD&SF does not have the requested asset cached and initiates a cache miss operation.
17. The CD&SF creates the Origin URI from the E6 URI by extracting the OriginContentId from the path of the E6 URI and adding to it the http scheme. The CD&SF passes the Origin URI to the Content Origin Function to request the asset as part of a cache fill operation.

    **Origin URI = http:// origin.sp.com/movies/hd/big.m2ts**

# 8 CONTENT DELIVERY AND USER OPERATION DURING PRESENTATION

From Figure 6, the CoD service permits several options for where the asset instance may be presented to the user in one or more temporal modes:

- In real time
- Progressive view
- Delayed view

During the presentation of the asset, the CoD service *may* provide "trick mode" capabilities (e.g., pause, fast forward, rewind). The mechanisms used for trick mode capabilities may be different depending on the temporal presentation method. Where trick mode functionality requires network support, the content control signaling would pass over the E6 reference point. Detailed specification of trick mode mechanisms is out of the scope of this specification.

During the presentation of the asset, the CoD service *may* permit the user, through the ITF, to create a "bookmark" to identify a particular point in time within the presentation of the asset. A bookmark may be used to identify where the playback may be restarted. Specification of the bookmark metadata and operations involving bookmarks are out of the scope of this specification.

The CoD service *shall* permit delivery of the content from the Service provider to an authorized user on any authorized ITF device at any authorized location. Many initial applications are expected to be based on fixed network infrastructure. Mechanisms to support location authorization are out of the scope of this specification. The further re-distribution of content beyond the ITF is out of the scope of this specification.

## 9 CONTENT ON DEMAND MEDIA FORMATS AND PROTOCOLS

This section specifies the media profiles formats, and protocols, media metadata, and media encapsulations for RTP and HTTP Delivery of CoD.

### 9.1 RTP MPEG-2 Transport Streams

The codecs and their encapsulation defined in the Linear TV specification (ATIS 0800018) Section 11, shall be supported except for the PiP profile (i.e., only SD and HD).

### 9.2 HTTP/HTTPS Clients and Servers

When either HTTP or HTTPS is used for interfaces, HTTP compliant request/response per ATIS-0800013, Section 7.6 (HTTP 1.1) *shall* be used. The server *shall* support all of the required (MUST) headers and *shall* adhere to all the required (MUST) procedures for a "server" defined in ATIS 0800013, Section 7.6 (HTTP 1.1). The client *shall* support all the required (MUST) headers and *shall* adhere to all the required (MUST) procedures for a "client" defined in ATIS 0800013, Section 7.6 (HTTP 1.1).

Some interfaces *may* support unique URI syntaxes, and they *may* explicitly define and support HTTP redirect functions when acting as content resolution servers.

Some interfaces *may* support HTTPS.

## 10 DETAILED REFERENCE POINTS SPECIFICATION

### 10.1 Introduction

This section provides detailed descriptions of key reference points identified in Figures 2 and 3 that are specific to the CoD Service specification.

### 10.2 Reference Point A3

The A3 reference point is between the Asset Preparation Functions and the CoD Application Function. This reference point is used by the CoD Application Function to request metadata created and stored in the Asset Preparation Functions.

The following functions are provided:

- list: get a list of assets
- get: get the metadata associated with an existing asset

A3 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the A3 URI as specified in section 7.4. Table 2 identifies the HTTP methods that are used on the A3 reference point. All requests originate on the CoD Application Function and terminate on the Asset Preparation Functions.

**Table 2: HTTP Methods for A3**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| list | GET | Path: ATIS-IIF-Assets | 200 OK | Body: list of A3 URIs |
| get | GET | Path: ATIS-IIF-Assets/CreatorAssetId (A3 URI) | 200 OK<br>404 not found | Body: metadata<br>(including asset encryption elements) |

### 10.3 Reference Point A8

The A8 reference point is between the IPTV Control Functions and the CoD Application Function. This reference point is used by the IPTV Control Functions during session setup to query the Application Function for authorization to deliver a requested asset to the ITF. If the subscriber associated with that request is authorized, the CoD Application Function returns the allocated session bandwidth and, optionally, the ECM and NPT playout position for that content if applicable. When a session is torn down, the IPTV Control Functions provide an update of the NPT playout position of the content for the subscriber using the A8 reference point.

The following functions are provided:

- authorization: check authorization for a subscriber to view a given asset
- NPT update: update NPT for an asset being view by a subscriber

A8 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the A8 URI as specified in section 7.8. Table 3 identifies the HTTP methods that are used on the A8 reference point. All requests originate on the IPTV Control Functions and terminate on the CoD Application Function.

**Table 3: HTTP Methods for A8**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| authorization | GET | Path: OriginContentId<br>Query: E3queryor S5query<br>Authorization: scheme** | 200 OK<br>401 unauthorized | Body: Allocated session bandwidth<br>Body: ECM*<br>Body: NPT* |
| NPT update | PUT | Path: OriginContentId<br>Query: E3queryor S5query<br>Authorization: scheme**<br>Body: NPT | 200 OK<br>401 unauthorized | |

*Optional
**See section 5.5.

### 10.4 Reference Point C1

The C1 reference point is between the Content Origin Function and the CD&LCF. This reference point is used by the Content Origin Function to notify the CD&LCF of relevant information associated with an asset. This information consists of a distribution policy (i.e., whether an asset needs to be pre-positioned).

When an asset needs to be pre-positioned in the CD&SF ("push" scenario), the Content Origin Function must provide (in the metadata passed along to the CD&LCF) an indication that the asset needs to be pre-positioned. The CD&LCF must factor this indication in the distribution policy associated with the asset and trigger pre-positioning of the corresponding asset in the CD&SF.

The Content Origin Function can invoke the following C1 functions:

- create: create an information set for an asset (i.e., distribution policy information) on the CD&LCF, which *may* result in the creation of a cache entry on the CD&SF
- list: list information sets known by the CD&LCF
- get: get information associated with an asset entry on the CD&LCF, where the delivered metadata consists of the distribution policy and the state of the asset on the CD&SF
- delete: delete an information set on the CD&LCF and purge the asset from the CD&SF

The C1 reference point uses the PUT method to create and the DELETE method to remove information about an asset on the CD&LCF. The message body of the PUT method is sent to the URI that includes the OriginContentId of the asset on the Content Origin Function. The message body of the PUT contains metadata that contains a pre-positioning policy.

Section 7.2 describes how the CD&LCF can use the OriginContentId to derive the Origin URI to which the pre-positioning information applies. When the OriginContentId for an asset is translated to the Origin URI, the OriginId is interpreted as the http host of the Content Origin Function that serves the asset, and the ContentId is interpreted as an http abs_path for the asset.

The list function URI contains a query that is used to filter the results of the list of OriginContentIds that is returned. The "max" keyword specifies the maximum number of OriginContentIds that are to be returned. The optional "start" keyword specifies the OriginContentId after which the returned list should start.

C1 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the C1 URI as specified in section 7.5. Table 4 identifies the HTTP methods that are used on the C1 reference point. All requests originate on the Content Origin Function and terminate on the CD&SF.

**Table 4: HTTP Methods for C1**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| create | PUT | Path: ATIS-IIF-Assets/OriginContentId<br>Body: Distribution Policy | 201 created<br>202 accepted | |
| delete | DELETE | Path: ATIS-IIF-Assets/OriginContentId | 200 OK<br>404 not found | |
| list | GET | Path: ATIS-IIF-Assets<br>Query: filtering criteria | 200 OK<br>404 not found | Body: list of OriginContentIds |
| get | GET | Path: ATIS-IIF-Assets/OriginContentId | 200 OK<br>404 not found | Body: metadata |

## 10.5 Reference Point C2

The C2 reference point is between the Content Origin Function and the Content Receiving Function within the CD&SF. This reference point is used by the Content Receiving Function to retrieve content from the Content Origin Function. Note that this reference point may also be used between instances of CD&SF in geographically distributed functional architectures. Since this reference point may be used between instances of CD&SF, this section refers to the server side interface (Content Origin Function or CD&SF) as the Media Resource Server and the client side interface (CD&SF) as the Media Resource Client.

The C2 reference point includes the ability to deliver content as part of a content pre-positioning flow as directed by the Distribution Control Function within the CD&LCF or as part of a cache miss cycle from the CD&SF. A subset of C2 functionality *may* be used in combination with limited Content Origin Function capabilities referred to as the Base Content Origin Function. Additional C2 functionality, used in combination with expanded Content Origin Function capabilities referred to as the Extended Content Origin Function, *may* be used for real time content delivery in support of CD&SF cache misses.

Content served from the Content Origin Function *may* be composed of multiple associated resources (e.g., normal and trick play continuous media, index data), which are referred to as Media Resources. The collection of Media Resources associated with an asset is referred to as a Media Resource Set. A Media Resource Server *shall* support the ability to serve Media Resources.

C2 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the C2 URI as specified in section 7.2. Table 5 summarizes the HTTP method that is used on the C2 reference point. This method is more fully specified in the sections that follow. All requests originate on the CD&SF and terminate on the Content Origin Function.

**Table 5: HTTP Method for C2**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| download | GET | Host: OriginId<br>Path: ContentId | 200 OK<br>302 redirect<br>404 not found | Link Header: Media Resource Metadata URI<br>Body: requested content |

### 10.5.1 Base Content Origin Function

This section specifies the C2 reference point as it relates to the Base Content Origin Function.

#### 10.5.1.1 Adherence to RFC 2616

The C2 reference point uses the HTTP 1.1 GET method to retrieve content from the Content Origin Function for delivery to the CD&SF. No capabilities for the Media Resource Client and Media Resource Server beyond those defined in section 9.2 are required. More specifically, the Media Resource Server appears to the Media Resource Client as a server as defined in section 9.2. The Media Resource Client appears to the Media Resource Server as a client as defined in section 9.2. A Media Resource Server that supports the C2 reference point *may* support the ability to serve any resource type allowed in RFC 2616 [25].

#### 10.5.1.2 Link Header for Media Resource Metadata URI

The Media Resources included in a Media Resource Set are described and located using Media Resource Metadata. C2 uses the HTTP Link header as specified in RFC 5958 [30] to provide the URI of the Media Resource Metadata. The Media Resource Server *shall* return the Link header in the response to an HTTP C2 download request for any Media Resource. The Media Resource Metadata URI *may* be used by the Media Resource Client to retrieve the Media Resource Metadata. Note that the presence of a Link header in the response to an HTTP download request indicates to the Media Resource Client that the referenced resource is a Media Resource.

This specification reserves the relation type “http://www.iif.atis.com/c2-media-resource-metadata” to indicate that the target URI specifies the URI of the Media Resource Metadata. The context URI is the URI of the HTTP request.

The example below shows the format of a Link header returned by a Media Resource Server to refer to the Media Resource Metadata with the URI “http://www.sp.com/my-c2-media-resource-metadata”.

```
Link: <http://www.sp.com/my-c2-media-resource-metadata>;
rel="http://www.iif.atis.com/c2-media-resource-metadata”
```

#### 10.5.1.3 Media Resource Metadata

Media Resource Metadata *may* be retrieved (or referenced) by the Media Resource Client using the HTTP GET method. A Content Origin Function *shall* use the Link header described in section 10.5.1.2 to specify the URI of the Media Resource Metadata associated with the requested Media Resource. Media Resource Metadata uses an XML based schema and *may* include the following information associated with resources related to the requested content:

- URI for each Media Resource (e.g., normal play, trick play, index) in the Media Resource Set
- Type of each Media Resource (e.g., continuous media, index data, etc.)
- Profile information related to each Media Resource
  - Continuous Media Resource (e.g., bitrate, transport container)
  - Non-continuous Media Resource (e.g., index data format type)

A profile specification identifies the properties of the resources associated with a particular continuous media type or application class. A profile specification also defines the mapping of Media Resource Metadata to a particular application. The MPEG-2 Transport Stream (TS) profile specification is given in section 10.5.1.4.

The schema for Media Resource Metadata is provided in ATIS-0800043, *Content on Demand Metadata Schema and Metadata Transactions* [17].

#### 10.5.1.4 MPEG-2 Transport Stream Profile

The MPEG-2 TS profile is utilized for an ITF that consumes MPEG-2 TS continuous media. A Media Resource that adheres to the requirements of this section is called an MPEG-2 TS Media Resource. MPEG-2 TS Media Resources may be referenced by the ITF using either the RTSP or HTTP variants of E3 and E6.

A Base Content Origin *shall* serve all MPEG-2 TS continuous media as MPEG-2 TS Media Resources.

#### 10.5.1.4.1 MPEG-2 TS Media Resource Set Description

This section describes the components of an MPEG-2 TS Media Resource Set. Each component that is included in the Media Resource Set includes an XML schema that is included as part of the Media Resource Metadata.

#### 10.5.1.4.2 MPEG-2 TS Continuous Media

MPEG-2 TS continuous media is the Media Resource used for a normal play back operation. An MPEG-2 TS Media Resource Set *shall* include a reference to this media resource in the Media Resource Metadata.

#### 10.5.1.4.3 MPEG-2 TS Trick Play Media Resource

An MPEG-2 TS trick play media resource may be used by the Media Resource Client to perform MPEG-2 TS trick mode operations. Each MPEG-2 TS trick play media resource represents a specific play speed that may be requested by the ITF. For ITFs that use RTSP as the session control protocol, the Media Resource Client may translate the value of the RTSP Scale header in an RTSP Play method to the appropriate trick mode speed referenced in the MPEG-2 TS Media Resource Set and stream the referenced resource to implement a trick mode operation.

An MPEG-2 TS Media Resource Set *may* reference one or more trick play media resources in the Media Resource Metadata.

Table 6 provides information about an MPEG-2 TS trick play media resource. The attribute in Table 6 is included as part of the Media Resource Metadata XML schema.

**Deleted:** Table 6

**Deleted:** Table 6

**Formatted:** Font: Book Antiqua

**Formatted:** Font: Book Antiqua

**Table 6: MPEG-2 TS Trick Play Media Resource Class**

| XML Value | Type | Values |
|---|---|---|
| @PlaySpeed.dlna.org | Integer | This attribute indicates the speed and direction of the asset. The 1x asset uses a value of 1. Tricks use negative numbers for reverse tricks, and forward numbers for forward tricks. For example, 4 represents the 4X forward trick, and -4 the 4X reverse or rewind trick. |

#### 10.5.1.4.4 MPEG-2 TS Index Data

MPEG-2 TS Index Data provides information on the attributes of the MPEG-2 TS continuous media, including the location of frames and in/out points within the MPEG-2 TS continuous media and the corresponding locations in the MPEG-2 TS trick play media resource(s).

The MPEG-2 TS Index Data may be used by the Media Resource Client to translate between time offsets used by an ITF and the byte offsets natively supported in HTTP. An ITF that uses RTSP as the session control protocol specifies time offsets using the RTSP Range header. The RTSP Range header specifies offsets in continuous media via NPT values. A Media Resource Client that supports RTSP-based streaming applications *may* use the MPEG-2 TS Index Data to translate between the NPT value provided in an RTSP Range header and byte offsets provided in an HTTP Range header.

An MPEG-2 TS Media Resource Set *should* include the MPEG-2 TS Index Data. Note that if a Media Resource Set does not include MPEG-2 TS Index Data, then the CD&SF might need to generate index data locally using the MPEG-2 TS continuous media resource.

Detailed specification of the MPEG-2 TS Index Data format is provided in section 10.5.4.

### 10.5.2 Extended Content Origin Function

For real time delivery in support of a CD&SF cache miss, it is necessary for the CD&SF to be receiving content from the Content Origin Function while delivering content to the ITF. In this case, special attention should be given not only to the prevention of CD&SF playout buffer under-runs (resulting from late arrival of content from the Content Origin Function), but also to subscriber play commands, including trick mode transitions.

The real time retrieval of continuous media *may* make use of two HTTP extensions. The first extends the HTTP range request to include time ranges based on the W3C Media Fragments URI specification [ref]. This allows a Media Resource Client to directly map RTSP stream control requests (that reference Normal Play Time) from an ITF to the corresponding HTTP download requests to a Media Resource Server. The second extension uses an ATIS IIF defined header (X-ATIS-IIF-ST-Profile) in the HTTP download request that allows the Media Resource Client to control the rate at which a continuous media resource is delivered. A Content Origin Function that supports these two C2 HTTP extensions is referred to as an Extended Content Origin Function.

#### 10.5.2.1 Adherence to Base Content Origin Function

The Extended Content Origin Function *shall* support all “*shall*” requirements of the Base Content Origin Function.

#### 10.5.2.2 Time Range Support

A Media Resource Server with Extended Content Origin Function *shall* support Time Ranges as specified in this section.

The Content Delivery Client Function within the ITF may use time as opposed to byte values to reference variable offsets (i.e., ranges) within continuous media resources that are ultimately served from the Content Origin Function. For example, the Content Delivery Client Function may use RTSP for stream control, where the range header in RTSP uses time as opposed to byte values to reference offsets within resources.

When the Content Delivery Client Function uses time values to reference offsets within a resource, the Media Resource Client may either translate the time offset into a byte offset locally using index data or pass the time offset to the Media Resource Server for resolution to byte offsets via the C2 reference point.

To support the two CD&SF behaviors described above, the C2 reference point supports time and byte offsets in continuous media resource requests. Byte offsets are supported natively in HTTP through the Range header.

Time ranges are supported using a syntax conformant to the W3C Media Fragments URI 1.0 specification: http://www.w3.org/TR/2010/WD-media-frags-20100624.

```
range-unit      =       bytes-unit | time-range-unit
```

```
       bytes-unit   =   "bytes"
  time-range-unit   =   “npt”
```

A new range specification header, called the time-ranges-specifier, is defined below. The time-ranges-specifier is a subset of the definition of the Range header from RFC 2326. The time-ranges-specifier specifies a range of time. The range is specified using npt units. Section 5.1.2 (Server Mapped Byte Ranges) of the Media Fragments specification specifies the syntax of a set of extensions to the HTTP Range header to support time-based ranges. This specification supports the Normal Play Time (NPT) time range option from section 5.1.2. Defined in RFC 2326 Section 3.6, NPT indicates the stream absolute position relative to the beginning of the presentation at the network level, and thus in particular does not take into account any frame reordering or network to decode or presentation delays. The timestamp consists of a decimal fraction. The part left of the decimal may be expressed in either seconds or hours, minutes, and seconds. The part right of the decimal point measures fractions of a second.

Time ranges are half-open intervals, including the lower point, but excluding the upper point. In other words, a range of a-b starts exactly at time a, but stops just before b. Only the start time of a media unit such as a video or audio frame is relevant. As an example, assume that video frames are generated every 40 ms. A range of 10.0-10.1 includes a video frame starting at 10.0 or later time and includes a video frame starting at 10.08, even though it lasted beyond the interval. A range of 10.0-10.08, on the other hand, would exclude the frame at 10.08.

When translating from NPT to byte offset, the Content Origin Function must consider acceptable inpoints and outpoints as defined in ANSI/SCTE 35 2007. For example, if the asset is an MPEG-2 TS carrying MPEG-2 video, the start of the response is expected to be the beginning of an I-frame. As a result, the starting offset may be slightly after, or the end of the response slightly before, the actual NPT specified in the request. The beginning of a continuous media resource corresponds to 0.0 seconds. Negative values are not defined.

The Range header is defined as follows:

```
                Range   =   "Range" ":" ranges-specifier
     ranges-specifier   =   byte-ranges-specifier | time-ranges-specifier
```

(Note that ranges-specifier is extended from RFC 2616 to include the time range specifier.)

```
time-ranges-specifier   =   ”t” “:” “npt” "=" npt-range-spec
       npt-range-spec   =   snpt "-" [enpt]
                 snpt   =   1*DIGIT; microseconds
                 enpt   =   1*DIGIT; microseconds
```

Here SNPT refers to the inpoint computed by the CD&SF. The Content Origin Function *shall* serve the named resource from a key frame closest to the SNPT value. The ENPT value denotes the outpoint at the end of the desired range. The Content Origin Function server *shall* serve the named resource up to a valid outpoint frame closest to the ENPT value. It is assumed that every continuous media resource starts with an NPT value of 0.

If the ENPT value is present, it *shall* be greater than or equal to the SNPT value, otherwise the npt-range-spec is invalid. The recipient of an npt-range-set that includes one or more invalid npt-range-spec values *shall* ignore the header field that includes that npt-range-set. If the ENPT value is absent, or if the value is greater than or equal to the total duration of the continuous media resource, ENPT is taken to be equal to one frame interval less than the total duration of the file in microseconds. By its choice of ENPT, a CD&SF can limit the number of frames retrieved without knowing the total duration of the continuous media resource.

If the npt-range-set cannot be satisfied, the Content Origin Function *shall* return a response with a status of 416 (requested range cannot be satisfied). Otherwise, the Content Origin Function *shall* return a response with a status of 206 (Partial Content) containing the satisfied ranges of the file.

A Base Content Origin Function is not required to understand a Time Range header and *may* therefore ignore a Range header that includes a time-based range-set. This is in sync with HTTP 1.1. This means that if a CD&SF provides a Time Range to a Base Content Origin Function, the complete resource *may* be delivered.

Assuming the Media Resource Server can map the given time range to a byte range, it will reply with an HTTP 206 Partial Content response.

#### 10.5.2.2.1 Time Range Response Header

Along with adding a new dimension for the HTTP Range request header, a new HTTP response header is introduced, called the Content-Range-Mapping, which provides the mapping of the retrieved byte range to the original time-based Range request. This header indicates the actual mapped range in terms of NPT values. This is necessary since the server might not be able to provide a range mapping that corresponds exactly to the requested range. Therefore, the Media Resource Client needs to be aware of this variance.

The specification for the Content-Range-Mapping header is based on the specification of the Content-Range header in RFC 2616 (section 14.16) and is shown below. Note that the Content-Range-Mapping header adds, in case of the temporal dimension, the instance start and end in terms of seconds after a slash "/" character in analogy to the Content-Range header. Also, an extension to the Accept-Ranges header (see RFC 2616, section 14.5) is introduced.

```
   Content-Range-Mapping   =   "Content-Range-Mapping" ":" ( content-range-mapping-spec
                               [ ";" "include-setup"] ) | "include-setup"
content-range-mapping-spec =   time-mapping-spec
        time-mapping-spec  =   “t” ":" npt-mapping-option
      npt-mapping-option   =   “npt” SP snpt   "-" enpt   "/"
                               [ snpt ] "-" [ enpt ]


           Accept-Ranges   =   "Accept-Ranges" ":" acceptable-ranges
       acceptable-ranges   =   1#range-unit *( "," 1#range-unit ) | "none"
```

(Note this does not represent the restriction that range-units can only appear once at most.)

```
              range-unit   =    bytes-unit | time-range-unit
              bytes-unit   =   "bytes"
         time-range-unit   =   “t”
```

The returned byte ranges *may* be used by the Media Resource Client to align portions of the full entity body, to re-generate a contiguous entity, if desired.

#### 10.5.2.3 MPEG-2 TS Index Data

An Extended Content Origin Function *shall* include the MPEG-2 TS Index Data in an MPEG-2 TS Media Resource Set.

#### 10.5.2.4 Scheduled Transmission Service

The entity body of a continuous media resource *may* be delivered to the Media Resource Client in real time using the scheduled transmission service. The scheduled transmission service may be used to deliver continuous media to the Media Resource Client using the specified content transfer parameters (e.g., rate). The scheduled transmission service ensures that continuous media is delivered using a requested rate. The scheduled transmission service also provides a burst capability. The burst capability may be used by the Media Resource Client to fill a buffer at a higher rate than the bit rate of the continuous media.

The scheduled transmission service is requested when the Media Resource Client issues an HTTP downloads request on a continuous media resource using an ATIS IIF-defined Scheduled Transmission (ST) Profile header (X-ATIS-IIF-C2-ST-Profile). Continuous media resources referenced by the Media Resource Client without the HTTP ST-Profile header *may* be delivered to the Media Resource Client using the default TCP behavior of the Media Resource Server.

The rate-multiplier parameter of the X-ATIS-IIF-C2-ST-Profile header is used by a Media Resource Client to specify the rate at which the continuous media is to be delivered to the client using the scheduled transmission service.

An Extended Content Origin Function *may* reserve resources at the server in order to deliver a continuous media resource requested by the Media Resource Client using the scheduled transmission service. The reservation *may* be used by the Media Resource Server to ensure that entity headers and entity data of the HTTP download response are delivered to the Media Resource Client using the ST profile specified in the X-ATIS-IIF-C2-ST-Profile header. The reservation is terminated by the Media Resource Server when either of the following two conditions is met:

1. The entity headers and entity data of the HTTP download response are delivered to the Media Resource Client.
2. The TCP connection between the Media Resource Client and Media Resource Server is terminated.

An Extended Content Origin Function that reserves resources *shall* support the ability to fail a Media Resource Client request due to lack of available resources. This specification uses error code 453 (Not Enough Bandwidth) in the HTTP download response for this purpose. RFC 2326 (RTSP) specifies the semantics of error code 453.

A Media Resource Server *shall* use the scheduled transmission service to return the entity body of a requested continuous media resource at the rate specified in the rate-multiplier parameter of the X-ATIS-IIF-C2-ST-Profile header when ALL of the following conditions are met:

1. The Media Resource Server supports the Extended Content Origin Function.
2. The Media Resource Client requests the scheduled transmission service via the HTTP X-ATIS-IIF-C2-ST-Profile header with requested rate of delivery provided using the rate-multiplier parameter.
3. The entity body of the requested resource contains continuous media.

A Base Content Origin Function is not required to understand the X-ATIS-IIF-C2-ST-Profile request header and *may* therefore ignore it. This is in sync with HTTP 1.1. This means that if a CD&SF provides the X-ATIS-IIF-C2-ST-Profile request header to a Base Content Origin Function, the Base Content Origin Function *may* deliver the resource using default TCP behavior.

The X-ATIS-IIF-C2-ST-Profile request header *shall* be provided in all C2 HTTP download requests that utilize the scheduled transmission service. As shown in Table 7, the X-ATIS-IIF-C2-ST-Profile request header provides support for five parameters: rate-multiplier, absolute-rate, start-time, burst-time and burst-rate-multiplier. The following sections specify the meaning of each of the parameters that *may* be included in the X-ATIS-IIF-C2-ST-Profile request header.

**Table 7: C2 HTTP Download Request Header for Scheduled Transmission Service**

| HTTP Header | Parameters | Units | Values |
|---|---|---|---|
| X-ATIS-IIF-C2-ST-Profile | rate-multiplier (optional),<br>absolute-rate (optional),<br>start-time (optional),<br>burst-time (optional),<br>burst-rate-multiplier (optional) | See parameter descriptions below. | See parameter descriptions below. |

### 10.5.2.4.1 Rate-Multiplier and Absolute-Rate

The X-ATIS-IIF-C2-ST-Profile request header *may* include either the rate-multiplier parameter or the absolute-rate parameter. The rate-multiplier and absolute-rate parameters specify the rate at which the Media Resource Client needs the continuous media resource delivered via the scheduled transmission service. The rate-multiplier parameter is specified as a multiplier on the bit rate of the continuous media as specified in the Media Resource Metadata. The absolute-rate parameter is specified as an absolute rate in bits per second. If neither the rate-multiplier or absolute-rate parameters are included in the X-ATIS-IIF-C2-ST-Profile request header, the rate by default will be 1X the bit rate of the continuous media as specified in the Media Resource Metadata.

The Media Resource Client *should* be prepared to receive the scheduled transmission at the requested rate. Failure to do so may result in the client's TCP receive window periodically closing on the Media Resource Server, and an effective transmission rate lower than the requested rate.

X-ATIS-IIF-C2-ST-Profile: rate-multiplier = 1*DIGIT "." 1*DIGIT

X-ATIS-IIF-C2-ST-Profile: absolute-rate = 1*DIGIT

#### 10.5.2.4.2 Start-Time

The X-ATIS-IIF-C2-ST-Profile request header *may* include the start-time parameter. If the start-time parameter is included in the X-ATIS-IIF-C2-ST-Profile request header, it indicates that the start of this transfer *shall* be delayed by the time specified. In this case, the transfer *shall not* begin until the specified amount of time has elapsed. After the specified time has elapsed, the transfer *shall* begin at the rate determined by the rate-multiplier parameter present in the header. The start-time parameter is mutually exclusive with the burst profile parameters.

If the start-time parameter is not included, then a start-time value of zero is assumed. That is, the Media Resource Server *shall* begin the transfer immediately upon receiving the request.

A Media Resource Client may use the start-time parameter to schedule transfers from the Media Resource Server in advance of when it is prepared to receive the data associated with the transfer. The Media Resource Server *may* use future scheduling requests to optimize its server and network data scheduling in advance of when the data must begin delivery to the Media Resource Client. A Media Resource Server that does not support the ability to schedule transfers in the future *shall* return HTTP error 501 (Not Implemented) to any ST profile header that includes this parameter.

X-ATIS-IIF-C2-ST-Profile: start-time = 1*DIGIT

#### 10.5.2.4.3 Burst Profile

The X-ATIS-IIF-C2-ST-Profile request header *may* include the burst profile parameters. A Media Resource Client may benefit from a capability to request delivery at a higher rate during some initial period of the scheduled transmission. Such initial bursts may be accommodated by specifying both burst-time and burst-rate-multiplier in the X-ATIS-IIF-C2-ST-Profile request header.

The burst-time parameter provides a method of specifying the size of the burst based on an amount of time. A time value is chosen over a size since the parameters used to determine the amount of buffering required in the Media Resource Client (network jitter, etc) are typically specified in units of time.

The burst-rate-multiplier parameter specifies the maximum rate (as a multiple of the continuous media bit rate) at which the Media Resource Server *may* transfer the data during the initial burst period.

If either burst profile parameter is included, then both *shall* be included, and the start-time parameter *shall not* be included. Start-time and burst profile are mutually exclusive.

The Media Resource Client *should* be prepared to receive the initial burst at the maximum requested burst rate. Failure to do so may result in the client's TCP receive window periodically closing on the Media Resource Server, and an effective delivery rate slower than the requested rate.

burst-rate-multiplier = 1*DIGIT "." DIGIT

> The maximum burst rate allowed for the burst. The rate is specified as a multiplier on the bit rate of the continuous media as specified in the Media Resource Metadata.

burst-time = 1*DIGIT

> The size of the burst represented as an amount of time. The time is specified in milliseconds.
>
> The size of the burst in bits may be obtained by determining the number of bits required to deliver the continuous media at the bit rate of the continuous media (as specified in

the Media Resource Metadata) over the period specified in the burst-time from the starting point specified in the Range header.

### 10.5.3 C2 Reference Point Use Case Examples

#### 10.5.3.1 Base Content Origin Function MPEG-2 TS Use Case Examples

This section includes flows that illustrate the use of the MPEG-2 TS profile for pre-positioning as well as cache miss using both the HTTP and RTSP variants of E3/E6.

##### 10.5.3.1.1 Prepositioning

Figure 27 shows an example C2 message sequence flow between a Media Resource Client and a Media Resource Server in support of the prepositioning flow in Figure 7 for an MPEG-2 TS Media Resource.

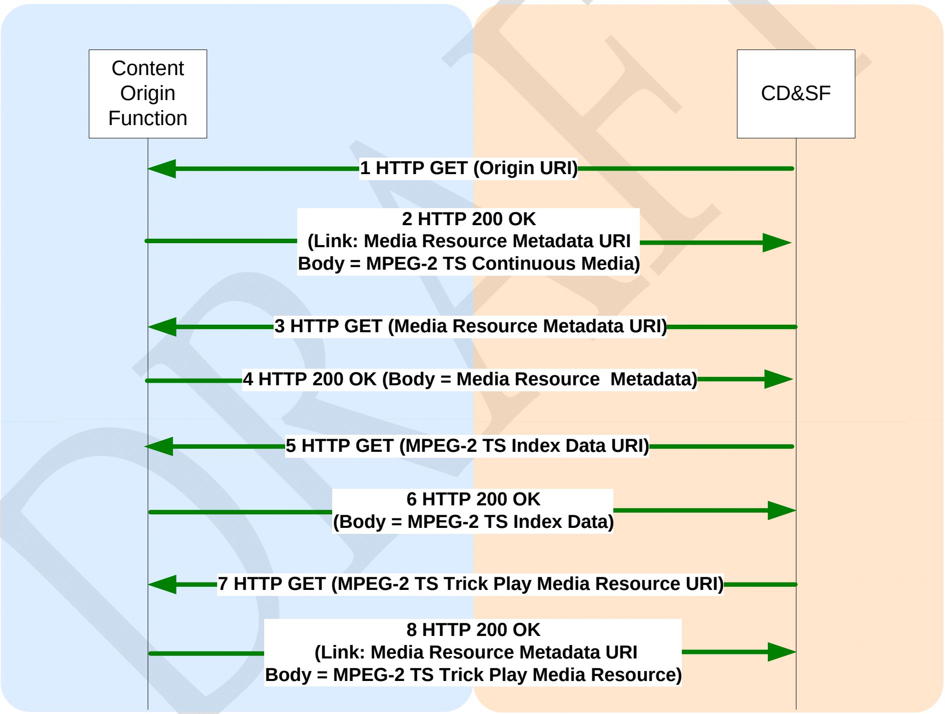

Content Origin Function
CD&SF
1 HTTP GET (Origin URI)
2 HTTP 200 OK
(Link: Media Resource Metadata URI
Body = MPEG-2 TS Continuous Media)
3 HTTP GET (Media Resource Metadata URI)
4 HTTP 200 OK (Body = Media Resource Metadata)
5 HTTP GET (MPEG-2 TS Index Data URI)
6 HTTP 200 OK
(Body = MPEG-2 TS Index Data)
7 HTTP GET (MPEG-2 TS Trick Play Media Resource URI)
8 HTTP 200 OK
(Link: Media Resource Metadata URI
Body = MPEG-2 TS Trick Play Media Resource)

**Figure 27: MPEG-2 TS Media Resource Prepositioning Flow**

1. The Media Resource Client uses the Origin URI to retrieve the MPEG-2 TS continuous media from the Media Resource Server.
2. The Media Resource Server returns the MPEG-2 TS continuous media to the Media Resource Client. Included in the response from the Media Resource Server is the Link header containing the Media Resource Metadata URI.

3. The Media Resource Client uses the Media Resource Metadata URI to retrieve the Media Resource Metadata from the Media Resource Server.
4. The Media Resource Server returns the Media Resource Metadata.

   The Media Resource Client uses the Media Resource Metadata to find the URI of a required Index Data resource.
5. The Media Resource Client retrieves the MPEG-2 TS Index Data resource using the HTTP GET method.
6. The Media Resource Server returns the MPEG-2 TS Index Data resource.
7. If included in the Media Resource Set, the Media Resource Client retrieves the MPEG-2 TS trick play media resource(s) using the HTTP GET method.
8. The Media Resource Server returns the trick play media resource.

#### 10.5.3.1.2 RTSP Initial Cache Miss

Figure 28 shows an example C2 message sequence flow between a Media Resource Client and a Media Resource Server in support of the cache miss shown in Figures 10 and 15 for an MPEG-2 TS Media Resource.

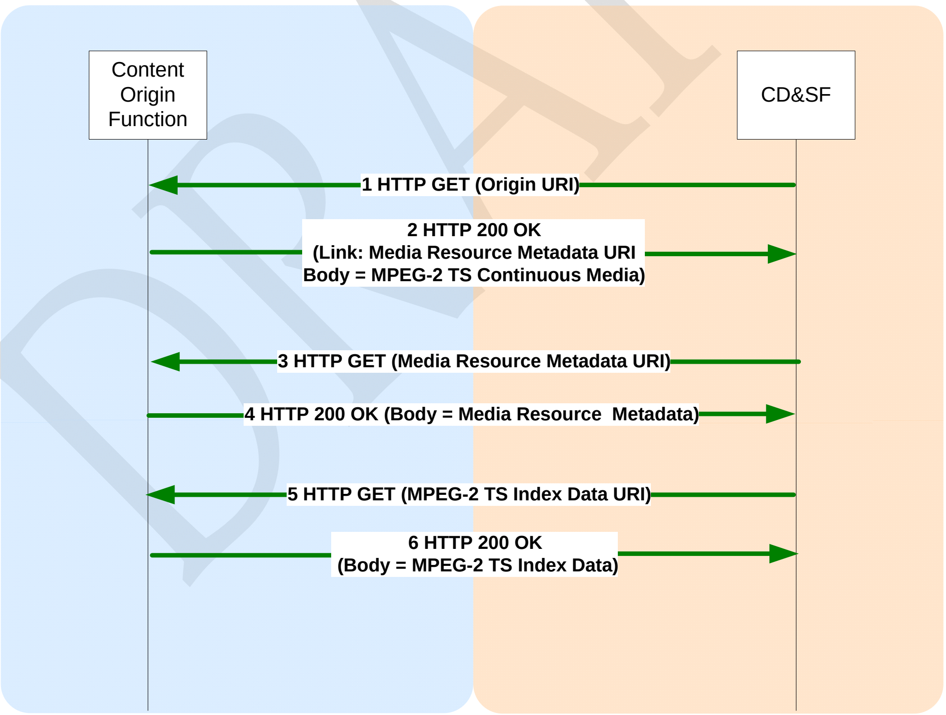

Content Origin Function
CD&SF
1 HTTP GET (Origin URI)
2 HTTP 200 OK
(Link: Media Resource Metadata URI
Body = MPEG-2 TS Continuous Media)
3 HTTP GET (Media Resource Metadata URI)
4 HTTP 200 OK (Body = Media Resource Metadata)
5 HTTP GET (MPEG-2 TS Index Data URI)
6 HTTP 200 OK
(Body = MPEG-2 TS Index Data)

**Figure 28: MPEG-2 TS Media Resource RTSP Initial Cache Miss**

1. The Media Resource Client uses the Origin URI to retrieve the MPEG-2 TS Continuous Media from the Media Resource Server.
2. The Media Resource Server returns the MPEG-2 TS continuous media resource to the Media Resource Client.

   Included in the response from the Media Resource Server is the Link header containing the Media Resource Metadata URI. If only the Media Resource Metadata URI is required, the Media Resource Client *may* use the HTTP HEAD method on the Origin URI in step 1 to retrieve the Media Resource Metadata URI without the continuous media.
3. The Media Resource Client uses the Media Resource Metadata URI to retrieve the Media Resource Metadata from the Media Resource Server.
4. The Media Resource Server returns the Media Resource Metadata.

   The Media Resource Client *may* cache the Media Resource Metadata returned by the Media Resource Server.

   The Media Resource Client uses the Media Resource Metadata to find the URI of a required Index Data resource.
5. The Media Resource Client *may* pre-fetch the MPEG-2 TS Index Data resource in support of future time-to-byte range requests such as with the RTSP trick mode cache miss flow illustrated below. This flow illustrates the Media Resource Client retrieving and caching the MPEG-2 TS Index Data resource using the HTTP GET method. If the Media Resource Metadata does not include an Index Data resource, then the Media Resource Client might generate index data locally from the MPEG-2 TS continuous media resource.
6. The Media Resource Server returns the MPEG-2 TS Index Data resource.

#### 10.5.3.1.3 RTSP Trick Mode Cache Miss with Cached Index Data

The following example C2 message sequence flow illustrates the use of the C2 reference point for a cache miss associated with step 5 of the RTSP Trick Mode Play flow shown in Figure 11 using the byte-based Range header natively supported in HTTP. The use case documented in this section assumes that the RTSP PLAY Method used to resume playback in step 5 of Figure 11 includes a Scale header with a value of 2 (2X Fast Forward). The NPT play position at the time of the RTSP PAUSE in step 2 is assumed to be 100.0 seconds.

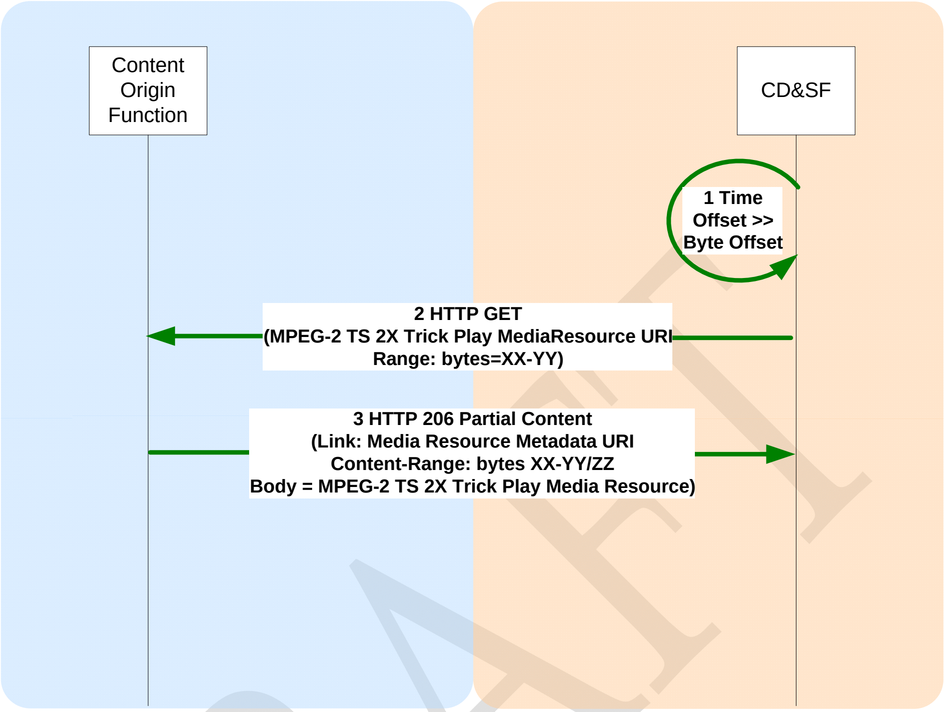

Content Origin Function
CD&SF
1 Time Offset >> Byte Offset
2 HTTP GET
(MPEG-2 TS 2X Trick Play MediaResource URI
Range: bytes=XX-YY)
3 HTTP 206 Partial Content
(Link: Media Resource Metadata URI
Content-Range: bytes XX-YY/ZZ
Body = MPEG-2 TS 2X Trick Play Media Resource)

**Figure 29: RTSP Trick Mode Cache Miss with Cached Index Data**

In this flow, the Media Resource Client progresses through the following steps:

1. The Media Resource Client uses the cached Index Data to translate the requested time offset into a byte offset within the trick play media resource.

   The Index Data resource contains information such as the byte offsets of the I, B, and P frames within the continuous media resource and within the associated trick play media resources. The Media Resource Client *may* use this information to translate the Time Range from the RTSP Range header to an HTTP byte range.
2. The Media Resource Client requests the resulting byte range from the Media Resource Server.
3. The Media Resource Client receives the 2X trick play media resource and forwards it to the ITF.

   The Media Resource Client *may* cache the 2X trick play media resource.

#### 10.5.3.1.4 HTTP Cache Miss

Figure 30 shows an example C2 message sequence flow between a Media Resource Client and a Media Resource Server in support of the cache miss shown in Figures 19 and 20 for an MPEG-2 TS Media Resource.

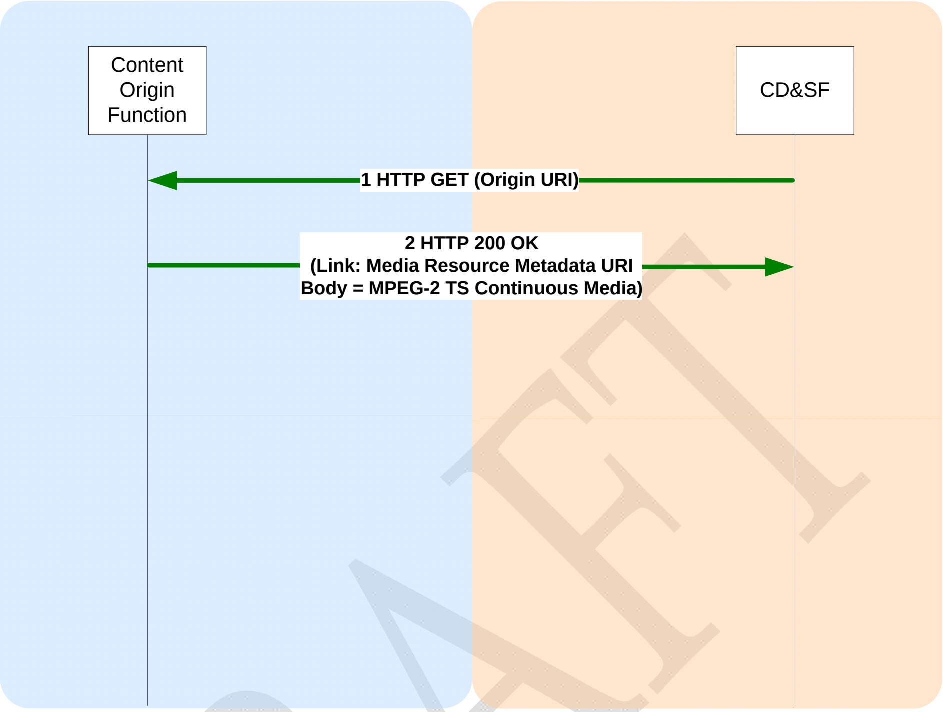

Content Origin Function
CD&SF
1 HTTP GET (Origin URI)
2 HTTP 200 OK
(Link: Media Resource Metadata URI
Body = MPEG-2 TS Continuous Media)

**Figure 30: MPEG-2 TS Media Resource HTTP Cache Miss**

1. The Media Resource Client uses the Origin URI to retrieve the MPEG-2 TS continuous media from the Media Resource Server.
2. The Media Resource Server returns the MPEG-2 TS continuous media to the Media Resource Client.

   Included in the response from the Media Resource Server is the Link header containing the Media Resource Metadata URI. If only the Media Resource Metadata URI is required, the Media Resource Client *may* use the HTTP HEAD method on the Origin URI in step 1 to retrieve the Media Resource Metadata URI without the continuous media.

### 10.5.3.2 Extended Content Origin Function MPEG-2 TS Use Case Examples

This section includes flows that illustrate the use of the MPEG-2 TS profile for the Extended Content Origin Function.

#### 10.5.3.2.1 RTSP Trick Mode Cache Miss with Time Ranges

The following example C2 message sequence flow illustrates the use of the C2 reference point for a cache miss associated with step 5 of the RTSP Trick Mode Play flow shown in Figure 11 using the time-base Range header. The use case documented in this section assumes that the RTSP PLAY method used to resume playback in step 5 of Figure 11 includes a Scale header with a value of 2 (2X Fast

Forward). The NPT play position at the time of the RTSP PAUSE in step 2 is assumed to be 100.0 seconds.

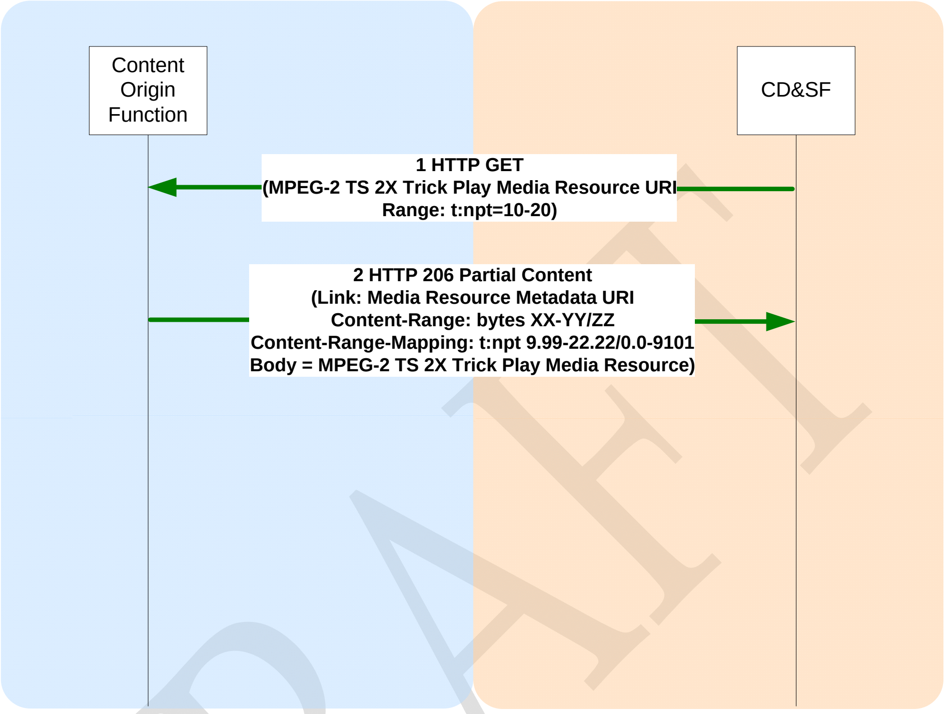

Content Origin Function
CD&SF
1 HTTP GET
(MPEG-2 TS 2X Trick Play Media Resource URI
Range: t:npt=10-20)
2 HTTP 206 Partial Content
(Link: Media Resource Metadata URI
Content-Range: bytes XX-YY/ZZ
Content-Range-Mapping: t:npt 9.99-22.22/0.0-9101
Body = MPEG-2 TS 2X Trick Play Media Resource)

**Figure 31: RTSP Trick Mode Cache Miss with Time Ranges**

1. The Media Resource Client requests the time range specified in the RTSP Range header to the Media Resource Server.

   The Media Resource Client maps the time range in the RTSP Range header to an HTTP time range. The Media Resource Server uses index data to map the requested time range to a byte range.
2. The Media Resource Client receives the 2X trick play media resource and forwards it to the ITF.

   The Media Resource Server returns the time and byte ranges resulting from the time range request in step 1 in the Content Range Mapping header.

   The Media Resource Client may cache the 2X trick play media resource.

#### 10.5.3.2.2 Example Cache Miss Flow with Scheduled Transmission Service

The following example illustrates the use of the X-ATIS-IIF-C2-ST-Profile header to successfully request the use of the scheduled transmission service to deliver a continuous media resource from the Content Origin Function to the CD&SF.

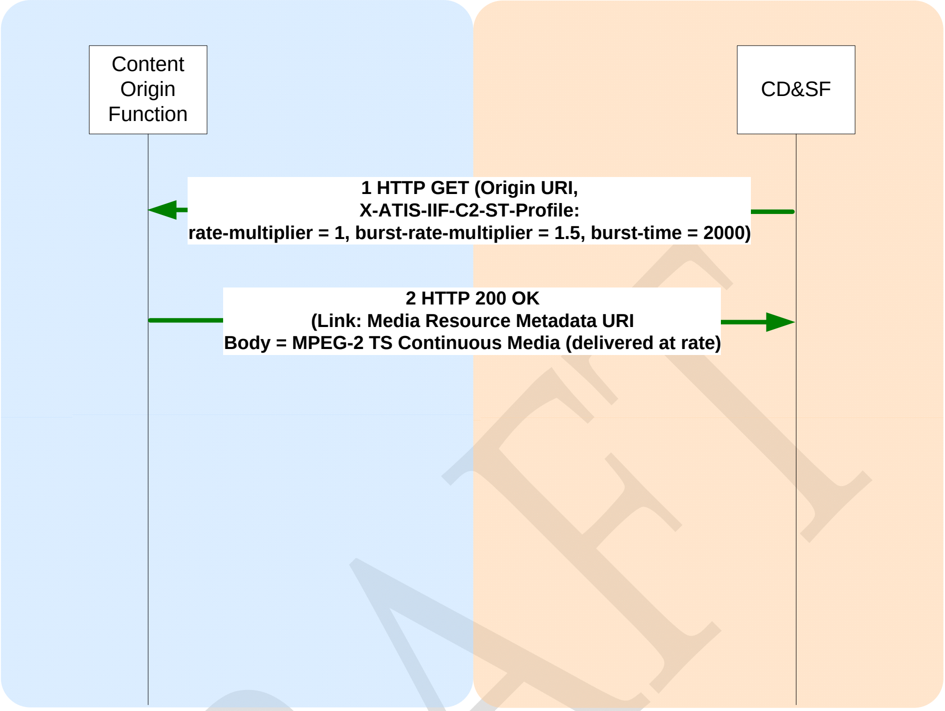

Content Origin Function
CD&SF
1 HTTP GET (Origin URI,
X-ATIS-IIF-C2-ST-Profile:
rate-multiplier = 1, burst-rate-multiplier = 1.5, burst-time = 2000)
2 HTTP 200 OK
(Link: Media Resource Metadata URI
Body = MPEG-2 TS Continuous Media (delivered at rate)

**Figure 32: Cache Miss Flow with Scheduled Transmission Service**

1. The Media Resource Client uses the Origin URI to retrieve the MPEG-2 TS continuous media from the Media Resource Server at a rate of 1X the bit rate specified in the Media Resource Metadata.

   The Media Resource Client requests that the transfer have a burst-time of 2 seconds (2000 milliseconds) and a burst-rate-multiplier of 1.5 times the bit rate of the continuous media as specified in the Media Resource Metadata.

   The X-ATIS-IIF-C2-ST-Profile header used for this request is:

   X-ATIS-IIF-C2-ST-Profile: rate-multiplier = 1, burst-rate-multiplier = 1.5, burst-time = 2000

2. The Media Resource Server returns the MPEG-2 TS continuous media to the Media Resource Client at the rate requested via the X-ATIS-IIF-C2-ST-Profile header.

   The transfer begins immediately (no delayed start) at the burst rate of 1.5 times the bit rate of the continuous media resource. While transferring at the 1.5 times the bit rate, the data is getting ahead by ½ second for each second that it is transmitting. After 4 seconds, the requested burst is delivered. At this point the transfer rate changes from the 1.5 multiplier specified in the burst-rate-multiplier parameter to the 1X rate multiplier specified in the rate-multiplier parameter. The transfer rate remains at the 1X rate-multiplier for the remainder of the transfer.

   Included in the response from the Media Resource Server is the Link header containing the Media Resource Metadata URI.

### 10.5.4 MPEG-2 TS Index Data Resource

This section describes the format of the entity body of the MPEG-2 TS Index Data resource.

Abbreviations:

- The bps abbreviation always means bits per second.
- The ppm abbreviation is parts per million.
- The term "transport packet" is used to represent the 188-byte "packets" that make up an MPEG-2 TS.
- TLV stands for Type, Length and Value. A TLV is a tuple, and is used to allow a compact and yet forward and backward-compatible representation of the data required.

#### 10.5.4.1 MPEG-2 TS Index Data Overview

The MPEG-2 TS Index Data resource contains metadata that is associated with MPEG-2 TS continuous media and MPEG-2 TS trick play media resources.

The MPEG-2 TS Index Data resource may be used by the Media Resource Client along with the MPEG-2 TS trick play media resource to implement trick mode operations.

The MPEG-2 TS Index Data resource includes information such as the locations of random access points associated with the MPEG-2 TS continuous media resource. Additional information is included to enable seamless splices between speeds. For example, the PMT is recorded so that the Media Resource Client can determine the PID to use in generating a PCR discontinuity on a transition to or from a trick. These parameters are provided to enable the Media Resource Client to generate such transitions if required in a particular application without having to parse a substantial portion of the stream in applications where doing so is suboptimal.

To stop a stream and switch speeds, the streaming application may need to stop at a clean Exit Point for the ITF decoder, if it requires such clean transitions. This means that the Media Resource Client may want to stop after one frame and before another, so the ITF won't display any partial frame on screen during the transition. Similarly, the Media Resource Client may want to stop before certain IDR, I or P frames because of frame reordering. In this way, the MPEG-2 TS Index Data resource keeps track of legal Exit Points in the MPEG-2 TS continuous media if this is required.

To switch speeds and switch to another MPEG-2 TS media resource, the Media Resource Client should receive data starting at an SRAP (SCTE Random Access Point, as defined in ANSI/SCTE 128 2010), so the Media Resource Client should know the SRAP locations. MPEG-2 TS Index Data resource keeps track of the legal Entry Points (SRAPs).

To switch streams, the Media Resource Client may need to generate a time base discontinuity, which means the Media Resource Client should insert a packet with a discontinuity bit set before switching to the new stream. This packet should have a PCR value that is roughly correct for that new stream at that point. Since the Media Resource Client may want to compute this rather than parsing the stream, this PCR value may be computed from the MPEG-2 TS Index Data resource. This means that the Media Resource Client should know the PCR value at the beginning of the MPEG-2 TS continuous media resource, an accurate estimate of the CBR bit rate and the offset where it will be inserting the PCR packet. Also, if the PCR values deviate significantly at any time from what might be expected, the Media Resource Client should track that in the MPEG-2 TS continuous media resource.

When the Media Resource Client switches streams, there is often a temporary frame gap, where nothing new is being fed to the streaming application to play. As this could be bothersome, the Media

Resource Client may need to be able to generate splices to play during the switch. To do this the Media Resource Client may need to know the frame size and the frame rate, both of which the Media Resource Client can obtain by examining the SPS header associated with the MPEG-2 TS continuous media resource.

The following sections describe how the Media Resource Client functionality described above is enabled by the MPEG-2 TS Index Data resource.

#### 10.5.4.2 MPEG-2 TS Index Data Format

An MPEG-2 TS Index Data resource can be logically broken into four sections—a signature, a header, a body and a trailer. Each of these will be described in more detail.

All defined TLVs use an 8-bit enum value for the type (values listed at the end of this section). TLVs with types equal to 64 or greater use a 16-bit length field. TLVs with types below 64 use an 8-bit length field for a more compact representation (these are generally TLVs in the body which are repeated many times). A TLV length is the length of the TLV not including the type or length fields. A TLV may be of length zero.

The type zero (0) is reserved. It cannot occur at the beginning of the MPEG-2 TS Index Data resource, but if otherwise present, it is simply a one-byte pad and can be used as needed to pad to any boundary desired.

TLVs do not need to begin on any particular byte value and are not aligned beyond the byte level.

Integers and unsigned integers should be assumed to be 4-byte values unless otherwise specified.

#### 10.5.4.3 MPEG-2 TS Index Data Signature

The MPEG-2 TS Index Data resource starts with several TLVs, called the signature, listed in this specific order:

<Signature TLV>

<Version TLV>

<End Section TLV>

<*Signature TLV*> is a TLV containing the ASCIIZ string "INDEX MPEG-2 TS" (without the quotes).

<*Version TLV*> is a TLV containing a pair of unsigned 16-bit values representing the major and minor revision with which this MPEG-2 TS Index Data resource complies. For this first version of the specification, the version will be 1.0 (major=1, minor=0). A major version increment can be assumed to be incompatible with previous versions (meaning code written to earlier versions). A minor version increment does not imply this, meaning that older implementations CAN interpret this file format, skipping over TLVs it does not understand or fields added at the end of existing TLVs it did not expect (so new fields can be added to the end of existing TLVs in some cases without breaking existing implementations). Generally, a new minor version is not required to introduce a new TLV, but the version is used just in case it is needed in the future.

<*End Section TLV*> contains the 32-bit data required to generate a checksum of 0x55555555 for the entire Signature, from the start of the Signature TLV to the end of the last byte of the End Section TLV.

#### 10.5.4.4 MPEG-2 TS Index Data Header

Other header TLVs follow the signature in the MPEG-2 TS Index Data header. In general, the order of these TLVs is not defined, unless otherwise specified. A given TLV may or may not be present in this section, unless otherwise specified. The Media Resource Server is expected to make a good faith effort to include any TLVs that make sense for this asset however.

TLVs that may be present in the MPEG-2 TS Index Data header include the following:

<*Vendor TLV*> is a TLV containing an ASCIIZ string representing the Media Resource Server vendor that ingested the MPEG-2 TS continuous media resource and generated this MPEG-2 TS Index Data resource. The value is either a three-byte IEEE Organizationally Unique Identifier (OUI) (lowest/oldest if the vendor has multiple OUI's assigned) assigned to that vendor, or a six-byte IEEE OUI-36/IAB as appropriate. It is included here to allow for a short private TLV, which may be used repeatedly during the body of the MPEG-2 TS Index Data.

<*Code Version TLV*> is a TLV containing an ASCIIZ string representing the version of the code on the Media Resource Server at the time the MPEG-2 TS continuous media resource was ingested. The format of this version is not defined. It is included here to allow for automated reingest of assets ingested with older code when a new version is installed, if required. It is also useful for diagnostic purposes.

<*Ingest Time TLV*> is a TLV containing an ASCIIZ string representing the ingest time locally on the Media Resource Server when this asset was ingested, in the format "YYYY-MM-DD HH:MM" where the values are zero-padded and HH is in 24-hour format. It is included here to allow for automated reingest based on date/time ranges if required. It is also useful for diagnostic purposes.

<*Bit Rate TLV*> specifies the bit rate of the MPEG-2 TS continuous media resource, in bits per second, as a 32-bit unsigned integer value. This value is expected to be computed, and may not be perfectly accurate, but every attempt should be made to make this accurate to within the required 50ppm. This TLV must be present.

<*PCR TLV*> specifies the PCR value at offset zero in the MPEG-2 TS continuous media resource. It contains a 32-bit unsigned transport packet offset (zero in this case) and a 64-bit unsigned integer PCR value. This is either present as a PCR in the stream at offset zero (meaning the first transport packet), or computed based on the bit rate and the first PCR. Care must be taken to account for rollover if it occurs, as this must be a legal PCR value. The transport packet offset is the offset in bytes from the beginning of the asset divided by the transport packet size (typically 188 bytes). Including this allows the Media Resource Client to generate PCR discontinuities if required on stream transitions without parsing the MPEG-2 TS continuous media resource.

<*PAT TLV*> specifies the complete PAT for the MPEG-2 TS continuous media resource, meaning the value is an entire transport packet specifying the PAT for this MPEG-2 TS continuous media resource. If the PAT occupies more than one transport packet in the asset, then multiple complete transport packets will be present here. This TLV must be present.

<*PMT TLV*> specifies the complete PMT for the MPEG-2 TS continuous media resource, meaning the value is an entire transport packet specifying the PMT for this MPEG-2 TS continuous media resource. If the PMT occupies more than one transport packet in the asset, then multiple complete transport packets will be present here. This TLV must be present. Including this TLV allows the Media Resource Client, if required, to determine the type of asset, any CODECs used, as well as other details without having to parse the MPEG-2 TS continuous media resource.

<*Elementary Stream Type TLV*> specifies the MPEG-2 Transport Stream elementary stream_type as defined in 13818-1 with Amendment 1:2007, table 2-34.

<*Image Format TLV*> contains the video image format information access point in the video elementary stream as defined by *Stream Type TLV*. For H.264, this is the SPS NAU header information (the entire byte stream, including the start codes), and for MPEG-2, this is the sequence header.

<*PTS TLV*> specifies the earliest PTS value for the specified elementary stream. The value contains first a 16-bit unsigned integer PID and then a 64-bit unsigned integer PTS value. Assuming there are multiple elementary streams in the asset, there would generally be one of these TLVs for each ES. For video, where frame reordering is possible, this would be the earliest PTS value, and not necessarily the first one.

<*Trick Speeds TLV*> specifies the trick speeds configured at the time of ingest on the Media Resource Server, and therefore the speeds of the created trick play media resources referenced in the MPEG-2 TS Index Data resource. The index of a given trick speed in this TLV is used in other places in this specification as the trick index. The 1x subfile is not listed in this TLV, and is assigned a trick index of zero. The first trick speed in this TLV is index 1. This is an array of signed 8-bit integers, so the maximum trick speed supported is ±127.

<*CC TLV*> is a TLV used to specify the initial Continuity Counter value for a given PID. The value contains first a 16-bit unsigned integer PID and then an 8-bit continuity counter value from the first transport packet of that PID in the stream. There is generally one of these for each PID in the MPEG-2 continuous media resource. These are included to allow the correct continuity counter value to be generated on any splices.

<*StreamId TLV*> is a TLV used to specify what the StreamId is for the primary video in the MPEG-2 continuous media, as defined in 13818-1. This is a one-byte unsigned integer value. If not present, it can be assumed that the StreamId is the usual 0xE0 value. This is included to allow the correct StreamId to be used in any frames in any splices generated if required.

<*ECM TLV*> is a TLV used to supply the initial ECM from the MPEG-2 TS continuous media resource, assuming it is encrypted and there are ECMs present in the asset when it was ingested. The value is the entire transport packet containing the first ECM control word in the asset. This is included to allow the ECM to be sent early if required to meet the minimum arm time for the encryption scheme.

<*Private TLV*> is a TLV that can be used to record something vendor specific or otherwise not defined in this version of the specification. A Private TLV contains first a three/six byte IEEE OUI or OUI-36/IAB for the vendor in question (as defined previously), then any additional information desired. Private TLVs should be ignored like any other TLVs if not understood.

<*End Section TLV*> is a special TLV that marks the end of the MPEG-2 TS Index Data header. It must appear last after all the other header TLVs. It must always be present. The value contains the 32-bit data required to generate a checksum of 0x55555555 for the entire header section, from the start to the end of the last byte of the End Section TLV. If there are any zero pad bytes between the end of the Signature and the first header TLV, they are not included in this calculation.

#### 10.5.4.5 MPEG-2 TS Index Data Body

The body of the MPEG-2 TS Index Data contains TLVs that primarily record the location of Entry and Exit Points in the MPEG-2 TS continuous media resource as well as in the MPEG-2 TS trick play media resources. However, it can also contain other information such as Elementary Stream Type or PCR TLVs if needed. The specific TLVs that are allowed within the body section are specified here.

This section primarily contains Entry Point TLVs and Exit Point TLVs for the MPEG-2 TS continuous media resource and Entry Point TLVs for the MPEG-2 TS trick play media resources. Entry Points are used for the translation of the start time in an RTSP range request (E6) to the start byte of an HTTP range request (C2). Exit Points are optionally used when the end time is requested in an RTSP range request for the translation of the end time in an RTSP range request (E6) to the end byte of an HTTP range request (C2) for normal play. For trick play, Entry/Exit Points (defined in Entry/Exit Trick TLVs) are used to translate both start and end times in RTSP range requests to byte offsets for HTTP range requests. Exit Points are optional and may be included in particular applications for which they are required. The MPEG-2 TS Index Data also may not contain all legal Entry and Exit Points if, in a particular application, it is determined that doing so is unnecessary. For example, it may only be necessary to note an Exit Point every 1/5th of a second to support user responsiveness goals in a particular application, rather than including all Exit Points.

Additional TLVs that may be contained in the body include the following TLVs mentioned previously:

<*PCR TLV*> specifies the PCR value at a given offset. Again, it can be computed. In a completely compliant CBR stream, there are no PCR TLVs in the body. However, if the PCRs that are calculated from the previous/initial PCR value and the bit rate are sufficiently inaccurate (as defined by 13818-1), then a PCR TLV should be generated to flag a new PCR value that can be used from this point forward. This can happen if the rate control of the encoder is not sufficiently accurate, there is a dropout in the encoder, there is some data loss in the file handling or for other reasons. The only change in this case is that the high-order bit of the first byte of the high-order byte of the 64-bit PCR value is set to “1” if the discontinuity bit in the asset is set, and zero if not.

<*Elementary Stream Type TLV*> See Header (section **Error! Reference source not found.**) for syntax and semantics.

<*Data Stream TLV*> contains indexes to TS packets of given PID and stream_type]. The TLV contains:

- A four-byte unsigned transport packet offset (offset 0 = start of file)
- A one-byte unsigned stream_type value for the indexed data stream packet
- The payload of the indexed data stream packet

<*Private TLV*> is a TLV that can be used to record something vendor specific or otherwise not defined in this version of the specification.

The body will consist primarily of the following TLVs:

<*Entry Point TLV*>, which contains:

- A four-byte unsigned transport packet offset (offset 0 = start of file) to the start of the Entry Point. This is the transport packet containing the PES header that defines the start of the SRAP.
- A two-byte unsigned frame length in transport packets for the entry point video frame
- A three-byte frame time, which is the number of frames times from the beginning of the asset.
- A one-byte flag specifying:
  4 bits (0xF0): The continuity counter from the PES transport packet
  1 bit (0x08): Set if the SRAP starts with an IDR
  3 bits (0x07): Reserved, set to zero

If an Entry Point TLV is followed by Entry/Exit Point Trick TLVs for one or more speeds in the other MPEG-2 TS trick play media resources, then those offsets correspond to the same logical point in the original asset. These must precede any other TLVs.

<Entry/Exit Point Trick TLV> which contains:

- A one-byte trick index
- A four-byte unsigned transport packet offset (offset 0 = start of file for forward tricks, or the end of the file for reverse tricks) to the start of the I-frame. This is the transport packet containing the PES header that defines the start of the SRAP.
- A one-byte flag specifying:
  4 bits (0xF0): The continuity counter from the PES transport packet
  1 bit (0x08): Set if the SRAP starts with an IDR
  3 bits (0x07): Reserved, set to zero

<Exit Point TLV> which contains:

- A four-byte unsigned transport packet offset. This is the transport packet containing the PES header for the frame AU that follows the Exit Point.
- A one-byte frame time delta, which is the (unsigned) number of frame times from the last I-frame in the 1x. If zero, the number is undefined.
- A one-byte flag specifying:
  4 bits (0xF0): The continuity counter from the PES transport packet
  3 bits (0x08): Reserved, set to zero

Exit Points are only recorded for the MPEG-2 TS continuous media resource and not for the MPEG-2 TS trick play media resources. Enough Exit Points should be recorded to support reasonable responsiveness when switching from the 1x stream as a result of a user request (pressing pause, rewind or fast-forward, for example). An Exit Point *should* be defined every 1/5th of a second.

<End Section TLV> once again appears at the end of the body. It must appear last after all the other body TLVs. It must always be present. The value contains the 32-bit data required to generate a checksum of 0x55555555 for the entire body section, from the start to the end of the last byte of the End Section TLV. If there are any zero pad bytes between the end of the header and the first body TLV, they are not included in this calculation.

#### 10.5.4.6 MPEG-2 TS Index Data Trailer

The MPEG-2 TS Index Data trailer includes a number of TLVs defining various elements that are not known until the MPEG-2 TS continuous media resource has been completely ingested. These include:

<*EOF TLV*> specifies the length of the MPEG-2 TS continuous media resource or MPEG-2 TS trick play media resource. It contains a one-byte trick index and a 16-bit unsigned integer transport packet count. A file containing one transport packet has a transport packet count of 1. This TLV is always present.

<*CC TLV*> specifies the final continuity counter for the specified PID. This is included to allow seamless splicing between assets in a playlist if necessary, including the correct continuity counter on any splice data that may be required in a particular application.

<*PTS TLV*> specifies the latest PTS value for the MPEG-2 TS continuous media resource video elementary stream. Again, if an asset ends with the sequence PBB, the P-frame is the last frame to be displayed and has the latest PTS.

<*End Section TLV*> once again appears at the end of the trailer.  It must appear last after all the other trailer TLVs.  It must always be present.  The value contains the 32-bit data required to generate a checksum of 0x55555555 for the entire trailer section, from the start to the end of the last byte of the End Section TLV.  If there are any zero pad bytes between the end of the body and the first trailer TLV, they are not included in this calculation.

#### 10.5.4.7 MPEG-2 TS Index Data Types

```
#define TYPE_UNUSED                              0
#define TYPE_ENTRY_POINT                         1
#define TYPE_TRICK_ENTRY_POINT                   2
#define TYPE_EXIT_POINT                          3
#define TYPE_SHORT_PRIVATE_START                 60
#define TYPE_SIGNATURE                           64
#define TYPE_INDEX_DATA_VERSION                  65
#define TYPE_END_SECTION                         66
#define TYPE_VENDOR                              67
#define TYPE_CODE_VERSION                        68
#define TYPE_INGEST_TIME                         69
#define TYPE_BIT_RATE                            71
#define TYPE_PCR                                 72
#define TYPE_PAT                                 73
#define TYPE_PMT                                 74
#define TYPE_SPS                                 75
#define TYPE_PTS                                 77
#define TYPE_TRICK_SPEEDS                        78
#define TYPE_CC                                  79
#define TYPE_STREAMID                            80
#define TYPE_ECM                                 81
#define TYPE_EOF                                 82
#define TYPE_ELEMENTARY_STREAM_TYPE              83
#define TYPE_IMAGE_FORMAT                        84
#define TYPE_DATA_STREAM                         85#define TYPE_PRIVATE
                                255
```

A number of private TLV enums are defined. Those below 64 allow some shorter TLVs to be recorded by the ingest vendor, and these TLVs are specific to the OUI specified in the <Vendor TLV>. The TLV specified by 255 can be used for any long form TLV and contains an OUI itself.

### 10.6 Reference Point C5

The C5 reference point is between the Asset Preparation Functions and the Content Origin Function. This reference point is used by the Asset Preparation Functions to notify the Content Origin Function of the availability of an asset (with a named OriginContentId) and its associated URI on the Asset Preparation Functions. This information may include the asset distribution policy (e.g., whether the asset needs to be pre-positioned).

The following functions are provided:

- create: create an information set for an asset (e.g., distribution policy information, C6 Asset Preparation Functions URI), which results in the creation of a resource on the Content Origin Function where the corresponding Origin URI is based on the OriginContentId as defined in section 7.2
- list: list information sets of assets known by the Content Origin Function
- get: get information associated with an asset entry, where the retuned metadata consists of the distribution policy and the state of the asset on the Content Origin Function
- delete: delete an information set and purge the asset from the Content Origin Function

The message body of a PUT to create an asset contains metadata that describes attributes of the asset and a C6 Asset Preparation Function URI from which the content is fetched via the C6 reference point.

The list function URI contains a query that is used to filter the results of the list of OriginContentIds that is returned. The "max" keyword specifies the maximum number of OriginContentIds that are to be returned. The optional "start" keyword specifies the OriginContentId after which the returned list should start.

C5 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the C5 URI as specified in section 7.3. Table 8 identifies the HTTP methods that are used on the C5 reference point. All requests originate on the Asset Preparation Functions and terminate on the Content Origin Function.

**Table 8: HTTP Methods for C5**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| create | PUT | Path: ATIS-IIF-Assets/OriginContentId<br>Body: C6 Asset Preparation Functions URI<br>Body: Distribution Policy | 201 created<br>202 accepted | |
| delete | DELETE | Path: ATIS-IIF-Assets/OriginContentId | 200 OK<br>404 not found | |
| list | GET | Path: ATIS-IIF-Assets<br>Query: filtering criteria | 200 OK<br>404 not found | Body: list of OriginContentIds |
| get | GET | Path: ATIS-IIF-Assets/OriginContentId | 200 OK<br>404 not found | Body: metadata |

## 10.7 Reference Point C6

The C6 reference point is between the Asset Preparation Functions and the Content Origin Function. This reference point is used by the Content Origin Function to retrieve content from the Asset Preparation Function.

The C6 reference point is identical to the Base Content Origin Function of the C2 reference point described in section 10.5.1. In this case, the Asset Preparation Functions appear as the Media Resource Server while the Content Origin Function appears as the Media Resource Client.

The Content Origin function *shall* determine whether a resource served by the Asset Preparation function is a Media Resource through the Link header described in section 10.5.1.2. The Content Origin Function *shall* use the prepositioning flow described in section 10.5.3.1.1 to pull Media Resources (as indicated by the Link header) from the Asset Preparation Function.

## 10.8 Reference Point E3

The E3 reference point is between the ITF Session Client Function and the IPTV Control Functions. This reference point is used to exchange session signaling information, which includes the OriginContentId of the requested content, between the ITF and the IPTV Control Functions.

### 10.8.1 Non-IMS-based CoD HTTP Content Delivery

For HTTP content delivery, the E3 reference point uses HTTP to request the delivery of content. E3 uses an HTTP transaction to trigger session setup in the network. The ITF may use the HTTP POST method to indicate the beginning of an HTTP-based streaming session.

The use of the POST method by the ITF to indicate the beginning of an HTTP-based streaming session is optional. Instead of using the POST method to indicate the beginning of an HTTP-based streaming session, an ITF may choose to request delivery of content via the HTTP GET method. The IPTV Service Control Function *shall* treat the HTTP POST and GET methods equivalently.

The following function is provided:

- setup: request content delivery

E3 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the E3 URI as specified in section 7.7. Table 9 identifies the HTTP methods that are used on the E3 reference point. All requests originate on the ITF and terminate on the IPTV Control Functions.

**Table 9: HTTP Method for E3**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| setup | POST,<br>GET | Path: OriginContentId<br>Authorization: scheme* | 302 redirect<br>401 unauthorized<br>404 not found | Location Header: E6URI |

*See section 5.5.

### 10.8.2 Non-IMS-based CoD Session Setup and Maintenance - RTSP

The RTSP SETUP method *shall* use the E3 URI as described in section 7.7.

In support of the Non-IMS Proxy CoD Session Establishment flow described in section 6.3, the IPTV Service Control Function returns RTSP 200 OK in response to the RTSP SETUP. The RTSP SETUP response includes headers for both a ControlSessionId and MediaSessionId. The RTSP TEARDOWN and ANNOUNCE methods *shall* use the E3 URI as described in section 7.7 without the query string.

The following functions are provided:

- setup: create a session
- teardown: delete a session
- event: announce and event

Table 10 identifies the RTSP methods that are used on the E3 reference point for the RTSP proxy use cases.

**Table 10: RTSP Proxy Methods for E3**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| setup | SETUP | Path: OriginContentId<br>Authorization : scheme*<br>Transport Header:<br>destination = ITF IP<br>client_port = ITF port | 200 OK<br>401 unauthorized<br>404 not found | Location Header: E6URI<br>Session Header: ControlSessionId<br>x-MediaSession Header: MediaSessionId |
| teardown | TEARDOWN | Path: OriginContentId<br>Session Header: ControlSessionId | 200 OK<br>404 not found | |
| event | ANNOUNCE | Path: OriginContentId<br>Session Header: ControlSessionId<br>Notice Header: notice-code | 200 OK | |

*See section 5.5.

In support of the Non-IMS Redirect CoD Session Establishment flow described in section 6.8, the IPTV Service Control Function returns an RTSP Temporary Redirect (status code 302) in response to the RTSP SETUP. The redirect response *shall* include the Location header with an E6 URI that is formed as described in section 7.11.

The following function is provided:

- setup: create a session

Table 11 identifies the RTSP methods that are used on the E3 reference point for the RTSP redirect use cases.

**Table 11: RTSP Redirect Methods for E3**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| setup | SETUP | Path: OriginContentId<br>Authorization: scheme*<br>Transport Header:<br>destination = ITF IP<br>client_port = ITF port | 302 redirect<br>401 unauthorized<br>404 not found | Location Header: E6URI |

*See section 5.5.

Table 12 identifies the RTSP methods that are used on the E3 interface, the compliance level requirements according to IETF RFC 2326, and the compliance-level requirements for E3.

**Table 12: RTSP Methods and Compliance for E3**

| RTSP Method | Direction:<br>C = Client (ITF)<br>S = Server (SCF) | IETF Compliance | E3 Compliance |
|---|---|---|---|
| ANNOUNCE | S→C | MAY | **SHALL** |
| DESCRIBE | C→S | MAY | MAY |
| SETUP | C→S | SHALL | SHALL |
| TEARDOWN | C→S | SHALL | SHALL |

If the IPTV Service Control Function is requesting termination of the RTSP session, the ANNOUNCE method *shall* be used. The reason for the termination *shall* be included in the Notice header of the ANNOUNCE message. The ANNOUNCE method *shall* be sent on the E3 interface only in the case where the IPTV Service Control Function is the entity requesting the termination of the session. The Notice-code and Notice-string pairs applicable to E3 are defined in Table 13.

**Table 13: RTSP Notice Codes for Termination Request for E3**

| Notice-code | Notice-string | Comment |
|---|---|---|
| 2401 | Ticket Expired | Viewing right expired. |
| 5200 | Server Resource Unavailable | Resource cannot be obtained. |
| 5401 | Downstream Failure | Stream could not be obtained. |
| 5402 | Client Session Terminated | |
| 5403 | Server Shutting Down | |
| 5404 | Internal Server Error | |

The ITF *should not* issue a SETUP request more than once for the same stream or multimedia session before issuing a TEARDOWN request.

The ITF *shall* provide a set of acceptable parameters in the transport header of the SETUP request. The IPTV Service Control Function *shall* respond with the selected parameters in the transport header of the RTSP 200 OK response. The transport headers *shall* conform to RFC 2326

Example Transport Header:

Transport: RTP/MP2T/UDP;unicast;destination=10.1.2.3;client_port=6970

### 10.8.3 IMS-based CoD Session Setup and Maintenance

The CoD SIP session shall be set up using TS 3GPP 24.229 as specified in ATIS-0800013 section 7.7 and the Session Description Protocol (SDP) as specified in ATIS-0800013 section 7.5. The use of SDP shall be between the ITF and the IMS core, and between the IMS core and the CD&SF. The control and communication from the IMS core to the NGN Transport Stratum components *shall* use the S3 reference point.

The SDP fields for a CoD SIP session shall include information related to the following:

- RTSP control channel for exchanging RTSP media control transactions between the ITF and the CD&SF
- Content delivery channel for streaming the media to the ITF

The SDP for the CoD SIP session shall be set in accordance with Table 14 below.

**Table 14: IMS SDP Normative Fields and Values for E3**

| SDP <type> | Description | <value> | Explanation |
|---|---|---|---|
| v= | protocol version | 0 | As specified in the standard referred to in ATIS-0800013 section 16. |
| s= | session name | "ATIS IIF CoD Service" | Identifies the session as being compliant to the IIF specifications for CoD Service. |
| i= | session description | Populated with the well known PSI for CoD Service including the OriginContentId | The Public Service Identifier (PSI) identifies that the session is a CoD service. |
| t= | time the session is active | Typically set to "0 0" | The t= type is usually defined as unbounded ("0 0"). |
| m= | media descriptions | <media> <port><br><proto> <fmt> | This field is mandatory and defines the media information for the RTSP control channel which uses TCP or TCP/TLS as per RFC 4145.<br>The<media> field is set to application.<br>The <port> value is set to the value 9 which is a discard port since the actual port that can receive the RTSP command is set dynamically and will be the one to be used by the peer.<br>The <proto> field is set to TCP or TCP/TLS.<br>The <fmt> parameter is set to IIF-iptv-rtsp. |
| a= | attributes | setup:active | As per RFC 4145. |
| a = | attributes | Connection: new | As per RFC 4145. |

| SDP <type> | Description | <value> | Explanation |
|---|---|---|---|
| c= | connection data | IN (IP4 orIP6 set appropriately) <connection address> | IN is defined by IANA as Internet. IP4 or IP6 refers to whether IP protocol version 4 or 6 is being used for the supplied connection address and should be set accordingly. The connection address is the IP address of the RTSP content control stream. |
| a= | attributes | <a=fmtp:>fmtp | There *may* be one or more of the lines that represent RTSP specific attributes (e.g., RTSP version).<br>a=fmtp:iptv_rtsp version <version number>. |
| m= | Additional media description entries | <media type> <port> <proto> <fmt> | Mandatory m= line that defines the video media. This line is constructed from the metadata associated with selected content by the user.<br>The<media> field is set to video<br>The <port> value is set to the IP port that can receive the video<br>The <proto> field is constructed from information from metadata.<br>The <fmt> parameter is set as follows:<br>- If MPEG2-TS is used, fmt is set to 33. |
| c= | connection data | IN (IP4 orIP6 set appropriately) <connection address> | IN is defined by IANA as Internet. IP4 or IP6 refers to whether IP protocol version 4 or 6 is being used for the supplied connection address and should be set accordingly. The connection address is the IP address where the media will be received. |
| b= | bandwidth information | AS: (session total bandwidth expressed in kilo bits per second) | The value is populated from the metadata. |
| b= | bandwidth information | RR: (Receiver Reports bandwidth expressed in kilo bits per second) | The value is populated by the ITF. If set to 0, then it indicates the ITF does not intend to send RTCP receiver reports (RR). A non-zero value indicates that the ITF intends to send RR reports. If RTCP reports are sent by an ITF, then they can be used as RTSP keep-alive messages.<br>If the ITF does not intend to send RTCP reports, then RTSP keep-alive messages *shall* be explicitly used. |
| a= | Attributes | recvonly | The ITF receives media only. |

#### 10.8.3.1 ITF Initiated SDP Offer during CoD IMS Session Setup

Upon a user request for a CoD session initiation, the ITF *shall* generate an initial INVITE request. The Request-URI in the INVITE request *shall* be set to the well-known PSI (Public Service Identifier) of the CoD Service including the OriginContentId for the selected content.

An initial SDP offer *shall* be included in the INVITE request. The SDP offer is composed in accordance with Table 14.

If the ITF receives a 488 error code with warning 370 Insufficient Bandwidth, the ITF *may* perform a new SIP INVITE with a lower maximum bandwidth for the CoD Service the ITF intends to view. This procedure may be repeated. If no agreement can be reached, the ITF *may* display a service unavailability message to the user.

When the ITF receives the SIP final response, the ITF *shall* validate that the answer complies with section 10.8.3.2 below, before displaying a message to the user to start streaming.

#### 10.8.3.2 CoD Service Control Functional Initiated Answer during IMS CoD Session Setup

Upon receipt of a SIP INVITE request, the CoD Service Control Function *shall* examine the Request-URI to determine that it is a CoD session initiation request, and *shall* validate the request against the user profile. Upon successful validation, the CoD Service Control Function attempts to establish an RTSP session with an appropriate CD&SF, using the SDP in the incoming SIP INVITE as a basis for that purpose. To establish an RTSP session, the CoD Service Control Function *shall* use an RTSP URI that has a path that is constructed by prepending the RTSP scheme to the <ContentInstanceIdentifier> component of the SIP URI.

If the RTSP session is successfully established, an RTSP 200 OK *shall* be received from the CD&SF. In turn, the CoD Service Control Function *shall* construct a SIP 200 OK that includes an SDP answer. The SDP answer *shall* include the following parameters, composed largely from the received RTSP 200 OK response.

- An m-line for the RTSP stream with the format m= <media type> <port> <proto> <fmt> and where:
    - The <media> *shall* have a value of “application”.
    - The <port> field *shall* be set to the port allocated for the RTSP stream.
    - The <transport> field *shall* be identical to the one in the initial INVITE.
    - The <fmt> field *shall* be identical to the one received 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 receiving RTSP commands (i.e. `c=IN IP4 <IP_ADDRESS>`)
- An a:=setup attribute *shall* be present and set to “passive”
- An a:=connection attribute *shall* be present and set to “new”
- One or more a:=fmtp lines representing specific RTSP attributes set as follows:
    - An “fmtp:iptv_rtsp h-uri” attribute *shall* be set to the RTSP URI to be used in 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 ITF by 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>). Note that an absolute URI *shall* have precedence over the c-line if the later is provided.
    - An “fmtp:iptv_rtsp h-session” attribute representing the session-id of the RTSP session to be used during media control. Optionally, a timeout parameter may be specified with a numeric timeout interval in seconds for keep-alive. If the timeout parameter is not specified, then a default value of 60 seconds *shall* be used (i.e. a=fmtp:iptv_rtsp h-session=<rtsp-session>[; timeout=<timeout>]).

Note that if RTCP is used, the media server can consider the reception of RTCP messages as indication of session “liveness,” and the server, in this case, should not expect to receive any explicit RTSP keep-alive messages.

- An m-line for the actual content  with the format m=  <media type> <port> <proto> <fmt> and where :
    - The <media> *shall* have a value of "video"
    - The <port> field *shall* be set according to allocated port
    - The <transport> field *shall* be identical to the one in the initial INVITE
    - The <fmt> field *shall* be present if one is included in the initial INVITE and *shall* be completed with the supported format by the CD&SF
- 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 for streamed media *shall* include the proposed session bandwidth.  This includes the IP and UDP headers. (ex. `b=AS:64 kbps`)
- A b-line for RTCP receiver reports.  If the offer included a zero value, then the answer *shall* be set to zero as well.  If the offer included a non-zero value, then the answer *shall* remain the same as to what is proposed.
- Optionally, a b-line for RTCP sender reports which *shall* specify the amount of bandwidth in kilobits per second allocated by the CD&SF for sending RTCP sender reports (ex. `b=RS:2 kbps`).
- An a-line with a "sendonly" (ex. `a=sendonly`)

The CoD Service Control Function does not perform any validation on the received answer from the CD&SF.  The answer is forwarded to the ITF.

#### 10.8.3.3 Well-Known CoD PSI

The Request URI line the INVITE request for a CoD service *shall* be a wild carded SIP PSI that conforms to the following:

SIP:\\IPTV-CoD-Service-<ContentInstanceIdentifier>@<*domain name*>

Where:

- IPTV-CoD-Service: is a well known component of the PSI, i.e., it is constant.
- Domain name is the service provider domain name obtained during service discovery.
- The ContentInstanceIdentifier is a wild card in the PSI.  The Content InstanceIdentifier is identical to the OriginContentId as described in section 7.1.

#### 10.8.3.4 IMS Session Parameters

The SubscriberId, DeviceId, and SuperCasId parameters are included in the SIP INVITE in the Authorization header according to RFC 3261.

### *10.9 Reference Point E6*

The E6 reference point is between the ITF Content Delivery Client Function and the CD&SF.  This reference point is used to exchange content control signaling information (e.g., play, pause, fast forward, rewind, download) between the ITF and the CD&SF.  This reference point corresponds to the E6 reference point in ITU-T Y.1910.

### 10.9.1 RTP Content Delivery

In support of the Non-IMS Proxy CoD Session Establishment flow described in section 6.3, an RTSP SETUP request is issued by the ITF to the IPTV Service Control Function via the E3 reference point and then proxied to the CD&SF via the S5 reference point. In response to the E3 RTSP SETUP request, the IPTV Service Control Function returns 200 OK to the ITF with headers containing the ControlSessionId, MediaSessionId and E6 URI (per Table 10). The E6 RTSP PLAY, PAUSE and ANNOUNCE methods *shall* use the returned E6 URI (as described in section 7.11 without the query string) and the MediaSessionId.

The following functions are provided:

- ♦ event: announce and event
- ♦ play: begin content delivery
- ♦ pause: pause content delivery

Table 15 identifies the RTSP methods that are used in the E6 RTSP proxy flows in this document.

**Table 15: RTSP Proxy Methods for E6**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| event | ANNOUNCE | Path: OriginContentId<br>Session Header: MediaSessionId<br>Notice Header: Notice-code | 200 OK | |
| play | PLAY | Path: OriginContentId<br>Session Header: MediaSessionId | 200 OK | |
| pause | PAUSE | Path: OriginContentId<br>Session Header: MediaSessionId | 200 OK | |

In support of the Non-IMS Redirect CoD Session Establishment flow described in section 6.8, an RTSP SETUP request is first issued by the ITF to the IPTV Service Control Function via the E3 reference point and in response, the IPTV Service Control Function returns an RTSP Temporary Redirect (status code 302). The Temporary Redirect includes the Location header with the E6 URI (per Table 11) that is formed as described in section 7.11. The ITF then re-issues an RTSP SETUP request to the CD&SF via the E6 reference point using the E6 URI. The E6 RTSP SETUP and subsequent PLAY, PAUSE and ANNOUNCE methods *shall* use the returned E6 URI as described in section 7.11.

The following functions are provided:

- ♦ setup: create a session
- ♦ teardown: delete a session
- ♦ event: announce and event
- ♦ play: begin content delivery at normal play mode
- ♦ pause: pause content delivery

Table 16 identifies the RTSP methods that are used in the E6 RTSP redirect flows in this document.

**Table 16: RTSP Redirect Methods for E6**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| setup | SETUP | Path: OriginContentId<br>Authorization: scheme*<br>Transport Header:<br>destination = ITF IP<br>client_port = ITF port | 200 OK<br>404 not found<br>401 unauthorized | Session Header: SessionId |
| teardown | TEARDOWN | Path: OriginContentId<br>Session Header: SessionId | 200 OK<br>404 not found | |
| event | ANNOUNCE | Path: OriginContentId<br>Session Header: SessionId<br>Notice Header: Notice-code | 200 OK | |
| play | PLAY | Path: OriginContentId<br>Session Header: SessionId | 200 OK | |
| pause | PAUSE | Path: OriginContentId<br>Session Header: SessionId | 200 OK | |

*See section 5.5.

Table 17 identifies the compliance level requirements according to IETF RFC 2326, and the compliance level requirements for E6.

**Table 17: RTSP Methods and Compliance for E6**

| RTSP Method | Direction:<br>C = Client (ITF)<br>S = Server (CD&SF) | IETF Compliance | E6 Compliance |
|---|---|---|---|
| ANNOUNCE | S→C | MAY | **SHALL** |
| PAUSE | C→S | SHOULD | **SHALL** |
| GET_PARAMETER | C→S | MAY | MAY |
| PLAY | C→S | SHALL | SHALL |
| SETUP | C→S | SHALL | SHALL |
| TEARDOWN | C→S | SHALL | SHALL |

The ANNOUNCE method *shall* be used by the CD&SF to inform the ITF that the RTSP session must be terminated and the reason for termination. The ANNOUNCE request *shall* include the Notice header. The Notice-code and Notice-string pairs applicable to E6 are defined in Table 18.

**Table 18: RTSP Notice Codes for E6**

| Notice-Code | Notice-String | Comment |
|---|---|---|
| 2101 | End-of-Stream Reached | |
| 2103 | Transition | The playing has caught up with a live stream |

| Notice-Code | Notice-String | Comment |
|---|---|---|
| 2104 | Start-of-Stream Reached | Can happen in case of rewind |
| 2105 | Stream Source Change | The source IP address of the media stream is changing. The headers “NewSource” and “NewSourceRetransmission” specify the new IP address and ports for the media stream(s). |
| 2401* | Ticket Expired | Viewing right expired |
| 4400 | Error Reading Content Data | Data read error |
| 5200* | Server Resource Unavailable | Resource cannot be obtained |
| 5401* | Downstream Failure | Stream could not be obtained |
| 5402* | Client Session Terminated | |
| 5403* | Server Shutting Down | |
| 5404* | Internal Server Error | |
| 6002 | Transition to next content | Playing next content in a playlist |

* Non-IMS RTSP Redirect Cases Only

The GET_PARAMETER method may be used to retrieve the CD&SF stream state and the NPT position within the stream. The GET_PARAMETER parameters that may be supported by the E6 interface are shown in Table 19. The ITF must not overload the CD&SF with too many GET_PARAMETER requests. The default request interval should be 1 minute.

**Table 19: RTSP GET_PARAMETER Parameters for E6**

| GET_PARAMETER Parameter | Result | Description |
|---|---|---|
| stream-state | current stream state | This parameter retrieves the current stream state. Possible returned values are:<br>playing<br>paused<br>stopped |
| position | NPT | This parameter retrieves the current time position in a CoD multimedia session. The position is the number of seconds from the beginning of the multimedia session in NPT format. This can be used for indication by the ITF to the user how far the presentation of the current session has advanced in time.<br>For example, the result of a GET_PARAMETER request with the parameter "position" can be:<br>position: npt=12:05:35.3- |

The ITF *should not* issue a SETUP request more than once for the same stream or multimedia session before issuing a TEARDOWN request.

The ITF *shall* provide a set of acceptable parameters in the transport header of the SETUP request. The CD&SF *shall* respond with the selected parameters in the transport header of the RTSP 200 OK response. The transport headers *shall* conform to RFC 2326

Example Transport Header:

Transport: RTP/MP2T/UDP;unicast;destination=10.1.2.3;client_port=6970

### 10.9.2 HTTP Content Delivery

The HTTP content delivery methods *shall* use the E6 URI as specified in section 7.11 and are based on HTTP 1.1 and HTTPS as specified in section 9.2. The E6 reference point uses HTTP transactions to trigger session setup and teardown in the network. The HTTP POST method *may* be used by the ITF for the session setup transaction.

When the HTTP POST method is received by the CD&SF, it *shall* signal the beginning of the HTTP session to the CoD Service Control Function via the S5 reference point. A CD&SF that successfully reserves resources associated with a session setup request *shall* return the assigned SessionId to the ITF in an HTTP Set-Cookie2 header conformant with RFC 2965. The 201 created response indicates successful creation of a session by the CD&SF. Note that the SessionId in the Cookie header is valid only for the resource specified in the URI of the POST method.

A CD&SF *should* limit the amount of time for which a session is held by setting a Max-Age value in the Set-Cookie2 header. The session *shall* expire when the amount of time specified in the Max-Age value has passed. The behavior of the timer associated with the Max-Age value is specified in RFC 2965 and RFC 2616.

The format of the Set-Cookie2 is specified as:

Set-Cookie2: ATIS-IIF-Session=<SessionId>; Max-Age=<value>; Version=1

The Location header in the response message *shall* contain the CD&SF Session URI used to identify the session resource established by the POST method. The CD&SF Session URI includes the assigned SessionId in the URI path. The CD&SF *shall* return the Location header in the response to a successful session setup request. The CD&SF Session URI *may* be used by the ITF to tear down the session.

The response message *may* include a bandwidth value. If the bandwidth value is included in the response, it tells the ITF how much bandwidth has been allocated for the session.

Once a session has been initiated by a CD&SF, an ITF *shall* include the SessionId, originally returned by the CD&SF in a Set-Cookie2 response header, in the Cookie header of any HTTP GET request that is associated with the session that was created by the POST method.

The format of the HTTP GET request Cookie header is specified as:

Cookie: $Version=1; ATIS-IIF-Session=<SessionId>

Note that the CD&SF *may* extend the Max-Age session expiration time by returning a new Max-Age value in an updated Set-Cookie2 header in any response.

A session is terminated by the CD&SF when the Max-Age value associated with the ATIS-IIF-Session Cookie has expired or when the ITF requests a teardown using the session teardown request implemented via the HTTP DELETE method on the E6 reference point. The DELETE method used to tear down a session *shall* use the CD&SF Session URI provided in the response to the POST method used to setup the session.

The format of the HTTP DELETE response Set-Cookie2 header is specified as:

Set-Cookie2: ATIS-IIF-Session=<SessionId>; Discard; Version=1

When the HTTP DELETE method is received by the CD&SF, it *shall* signal the end of HTTP session to the IPTV Control Functions via the S5 reference point. The “Discard” parameter in the Set-Cookie2 header of the response is an indication to the ITF that the cookie associated with the session *shall* be discarded.

An ITF may choose to not explicitly indicate the beginning of a video streaming session via the E6 HTTP POST method. In this case, the ITF uses the HTTP GET method to request content delivery without explicitly indicating the beginning of a video streaming session to the CD&SF. When the CD&SF receives an E6 HTTP GET request outside of the context of an HTTP session (i.e., there is no ATIS-IIF-Session Cookie in the GET request), the CD&SF *shall* use the S5 reference point to signal the beginning of the HTTP session to the CoD Service Control Function. As described above, the CD&SF *shall* include the Set-Cookie2 header in the HTTP 200 OK response to the ITF HTTP GET request. Subsequent ITF GET requests related to this HTTP session, if any, *shall* include the ATIS-IIF-Session Cookie header as described above. As there is now no explicit session delete message from the ITF to the CD&SF to signal the end of the HTTP session, the CD&SF *shall* use the S5 reference point to signal the end of the HTTP session when either the entire asset associated with the HTTP GET is delivered to the ITF or the Max-Age value for the ATIS-IIF-Session Cookie expires.

The following functions are provided:

- setup: create a session
- teardown: delete a session
- download: download content

Table 20 identifies the HTTP methods that are used on the E6 reference point. All requests originate on the ITF and terminate on the CD&SF.

**Table 20: HTTP Methods for E6**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| setup | POST | Path: OriginContentId<br>Authorization: scheme* | 201 created<br>302 redirect<br>401 unauthorized<br>404 not found | Set-Cookie2 Header: ATIS-IIF-Session=<SessionId>; Max-Age=<value>; Version=1<br>Location Header: CD&SF Session URI<br>Body: allocated session bandwidth |
| teardown | DELETE | URI: CD&SF Session URI | 200 OK<br>204 no content<br>404 not found | Set-Cookie2 Header: ATIS-IIF-Session=<SessionId>; Discard; Version=1 |
| download | GET | Path: OriginContentId<br>Authorization: scheme*<br>Cookie Header: $Version=1; ATIS-IIF-Session=<SessionId> | 200 OK<br>302 redirect<br>401 unauthorized<br>404 not found | Set-Cookie2 Header: ATIS-IIF-Session=<SessionId>; Max-Age=<value>; Version=1<br>Body: requested content |

*See section 5.5.

## *10.10 Reference Point S1*

The S1 reference point is between the CoD Service Control Function and the Location Control Function within the CD&LCF. This reference point is used by the CoD Service Control Function to locate an instance of the CD&SF capable of delivering the requested content to the ITF. The S1 reference point corresponds to the non-IMS S1 reference point in ITU-T Y.1910.

The ATIS-IIF-Subscriber-Scheme and ATIS-IIF-ITF-IP-Address are required for the CD&LCF to select an instance of a CD&SF most suitable for delivery to that ITF.

The following function is provided:

- locate: locate an instance of the CD&SF

S1 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods *shall* use the S1 URI as specified in section 7.9. Table 21 identifies the HTTP methods that are used on the S1 reference point. All requests originate on the IPTV Control Functions and terminate on the CD&LCF.

**Table 21: HTTP Methods for S1**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| locate | GET | Path: OriginContentId<br>Query: ATIS-IIF-Subscriber-Scheme<br>Query: ATIS-IIF-ITF-IP-Address | 200 OK<br>404 not found | Body: CD&SF-host |

## *10.11 Reference Point S5*

The S5 reference point is between the CoD Service Control Function and the CD&SF. This reference point is used to exchange session management information between the CoD Service Control Function and the CD&SF. The S5 reference point corresponds to the non-IMS S5 reference point in ITU-T Y.1910.

### 10.11.1 RTSP Proxy Session Setup

The RTSP SETUP request initiated by the CoD Service Control Function to the CD&SF *shall* use the S5 Proxy URI as specified in section 7.10.

In support of the Non-IMS Proxy CoD Session Establishment flow described in section 6.3, the CD&SF returns RTSP 200 OK in response to the RTSP SETUP request sent by the CoD Service Control Function. The RTSP 200 OK response includes a session header containing the MediaSessionId. The RTSP TEARDOWN and ANNOUNCE methods *shall* use the S5 Proxy URI as described in section 7.10 without the query string.

The following functions are provided:

- setup: create a session
- teardown: delete a session
- event: announce and event
- status: request status of a specified session

Table 22 identifies the RTSP methods that are used on the S5 reference point.

**Table 22: RTSP Methods for S5**

| Function Name | Method | Request Message Information | Status code | Response Message Information |
|---|---|---|---|---|
| setup | SETUP | Path: OriginContentId<br>Authorization: scheme*<br>Transport Header:<br>destination = ITF IP<br>client_port = ITF port | 200 OK<br>401 unauthorized<br>404 not found | Session Header: MediaSessionId<br>Transport Header:<br>destination = ITF IP<br>client_port = ITF port<br>source = CD&SF Ud IP<br>server_port = CD&SF Ud port<br>ATIS-IIF-content_bandwidth = bandwidth |
| teardown | TEARDOWN | Path: OriginContentId<br>Session Header: SessionId | 200 OK<br>404 not found | |
| event | ANNOUNCE | Path: OriginContentId<br>Session Header: SessionId Notice Header: Notice-code | 200 OK | |
| status | GET_PARAMETER | Path: OriginContentId<br>Session Header: SessionId<br>Body: position | 200 OK<br>404 not found | Body: NPT |

*See section 5.5.

Table 23 identifies the compliance level requirements according to IETF RFC 2326, and the compliance level requirements for S5.

**Table 23: RTSP Compliance for S5**

| RTSP Method | Direction:<br>C = Client (COD SCF)<br>S = Server (CD&SF) | IETF Compliance | S5 Compliance |
|---|---|---|---|
| ANNOUNCE | S→C | MAY | **SHALL** |
| SETUP | C→S | SHALL | SHALL |
| TEARDOWN | C→S | SHALL | SHALL |
| GET_PARAMETER | C→S | MAY | SHALL |

The SETUP method may contain the maximum allowable bandwidth in the request message. The CD&SF may return the allocated bandwidth used in the session.

NPTs are relative to the beginning of the content segment in which they occur.

The ANNOUNCE method *shall* be used by the CD&SF to inform the CoD Service Control Function that the RTSP session must be terminated and the reason for termination. The announcement uses S5 in combination with E3 in the non-IMS proxy CD&SF initiated session termination. The ANNOUNCE request *shall* include the Notice header. The Notice-code and Notice-string pairs applicable to S5 are defined in Table 24.

**Table 24: RTSP Notice-Codes for S5**

| Notice-Code | Notice-String | Comment |
|---|---|---|
| 2401 | Ticket Expired | Viewing right expired. |

| Notice-Code | Notice-String | Comment |
|---|---|---|
| 5200 | Server Resource Unavailable | Resource cannot be obtained. |
| 5401 | Downstream Failure | Stream could not be obtained. |
| 5402 | Client Session Terminated | |
| 5403 | Server Shutting Down | |
| 5404 | Internal Server Error | |

The CoD Service Control Function *should not* issue a SETUP request more than once for the same stream or multimedia session before issuing a TEARDOWN request.

The CoD Service Control Function *shall* provide a set of acceptable parameters in the transport header of the SETUP request. The CD&SF *shall* respond with the selected parameters in the transport header of the RTSP 200 OK response. The transport headers *shall* conform to RFC 2326

Example Transport Header in Request Message:

Transport: RTP/MP2T/UDP;unicast;destination=10.1.2.3;client_port=6970

Example Transport Header in Response Message:

Transport:RTP/MP2T/UDP;unicast;destination=10.1.2.3;client_port=6970;source=10.4.5.6;
server_port=3456;ATIS-IIF-content_bandwidth=8000000

**10.11.2 Redirect Session Setup**

For the redirect case, HTTP is used by the CD&SF to request access to the service and resources from the CoD Service Control Function. It is also used by the CoD Service Control Function to initiate termination of a session.

The following functions are provided:

- access: request service access (CD&SF to COD-SCF)
- release: release service access (CD&SF to COD-SCF)
- query: request status of a specified session (CD&SF to COD-SCF)
- terminate: inform the CD&SF that the session must be terminated (COD-SCF to CD&SF)
- list: request a list of active sessions (COD-SCF to CD&SF)
- status: request status of a specified session (COD-SCF to CD&SF)

For the service request message, the path of the POST contains the asset being requested and the query string contains data that identifies the ITF or user who is requesting the session and, optionally, resources that are needed by the session. This will typically be the same query string that was received by the CD&SF via the E6 download or setup request. The optional resource information includes information that may be used in an S3 reservation request such as a range of allowed bandwidth values required to deliver the asset (i.e., ATIS-IIF-MinBandwidth, ATIS-IIF-MaxBandwidth), the URI scheme received by the CD&SF via the E6 download or setup requests, and a specification for the flow sent on the Ud reference point. This scheme may be used by the CoD Service Control Function to determine which Ud encapsulation should be used for the session (HTTP vs RTP) and therefore which reservation class to request via the S3 interface. The Location header in the response message contains the COD-SCF Session URI used to identify the session resource established by the POST method. The COD-SCF

Session URI includes the assigned SessionId in the URI path. The response message optionally contains content protection information (e.g., ECM) and a maximum bit rate to which the CD&SF is expected to limit itself.

The service release message uses the COD-SCF Session URI, with the message body of the DELETE containing the reason code for the termination and the current NPT of the content being delivered.

The query message is used by the CD&SF to request status of an established session and includes in the query string data that identifies the ITF or user associated with the session. The response message optionally contains content protection information (e.g., ECM) and a maximum bit rate to which the CD&SF is expected to limit itself.

The terminate message is sent from the COD-SCF to the CD&SF to inform the CD&SF that a session must be terminated. The query string of the request indicates that this is a termination request and the message body of the response contains the current NPT of the content being delivered, if applicable.

S5 is based on HTTP 1.1 and HTTPS as specified in section 9.2. The HTTP methods for the CD&SF to COD-SCF requests *shall* use the S5 Redirect URI as specified in section 7.12. The HTTP methods for the COD-SCF to CD&SF requests *shall* use the COD-SCF Session URI as specified in section 7.12. Table 25 identifies the HTTP methods that are used on the S5 reference point. The requests associated with the first three listed functions originate on the CD&SF and terminate on the CoD Service Control Function, while the requests associated with the last three listed functions originate on the CoD Service Control Function and terminate on the CD&SF.

**Table 25: HTTP Methods for S5**

| Function Name | Method | Request Message Information | Status Code | Response Message Information |
|---|---|---|---|---|
| access | POST | Path: OriginContentId<br>Query: ATIS-IIF-MinBandwidth<br>Query: ATIS-IIF-MaxBandwidth<br>Query: ATIS-IIF-Subscriber-Scheme<br>Query: ATIS-IIF-SourceIP<br>Query:ATIS-IIF-SourcePort<br>Query: ATIS-IIF-DestIP<br>Query:ATIS-IIF-DestPort<br>Authorization: scheme*** | 201 created<br>401 unauthorized<br>453 not enough bandwidth | Location Header: COD-SCF Session URI<br>Body: allocated session bandwidth<br>Body: ECM* |
| release | DELETE | URI: COD-SCF Session URI<br>Body: Reason<br>Body: NPT | 200 OK<br>404 not found | |
| query | GET | URI: COD-SCF Session URI<br>Authorization: scheme*** | 200 OK<br>401 unauthorized<br>404 not found | Body: allocated session bandwidth<br>Body: ECM* |
| terminate** | POST | Path: ATIS-IIF-Sessions/SessionId<br>Query: ATIS-IIF-S5-Notify = <terminate> | 200 OK<br>404 not found | Body: NPT |
| list** | GET | Path: ATIS-IIF-Sessions | 200 OK<br>404 not found | Body: list of SessionIds |
| status** | GET | Path: ATIS-IIF-Sessions/SessionId | 200 OK<br>404 not found | Body: NPT |

*Optional

**Required for RTSP, optional for HTTP

***See section 5.5.

# APPENDIX A: CONTENT MANAGEMENT SYSTEM OVERVIEW (INFORMATIVE)

## *A.1 Overview*

The Content Management System (CMS) provides the ingestion and preparation of assets and facilitates flexible process flows to enable a unified ingestion point for all video platforms used to serve end customers.  This section describes a Content Management System architecture and its supporting network elements and interfaces.  It attempts to unify and integrate various services and applications into a consistent and strategic architecture and design approach in order to optimize network and system resources.

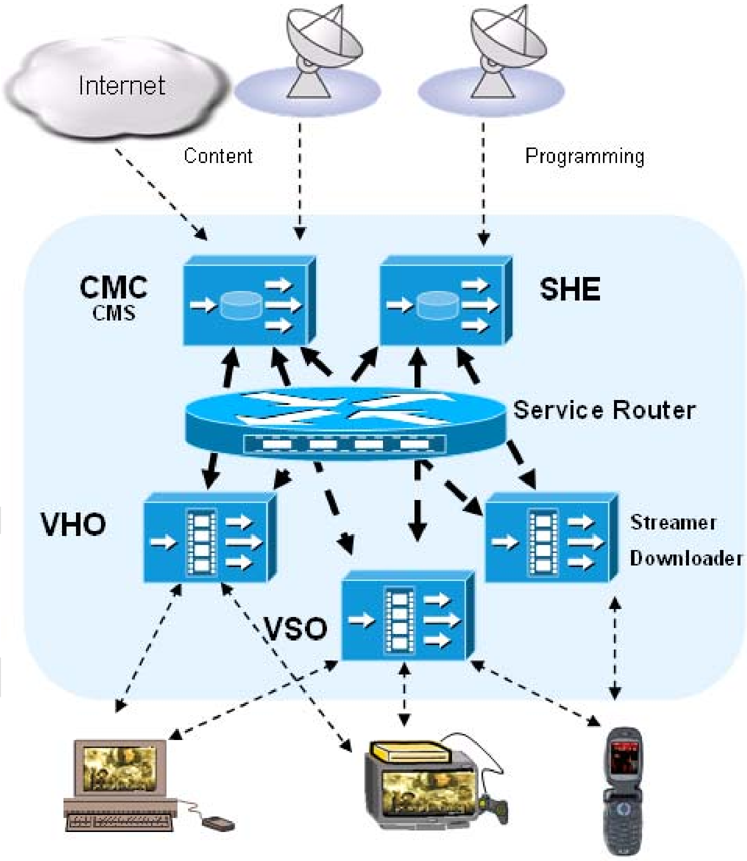


**Figure 33: Content Delivery Services**

The CMS supports on-demand interactive services, where assets are pre-processed and pre-staged prior to offering them to end users. Assets can be movies, TV shows, sporting events, games, advertising video clips, graphics, text and data contents including their corresponding description data and metadata files.

The CMS platform provides VoD and advertising content processing system units integrated with subsystems required to deliver these services. These system modules are functional design and logical components clustered together to deliver the required functionalities. The CMS system architecture, high-level design concept and network implementation requirements are described. The main design objectives and concerns in proposing this architecture are to provide interoperability with video interactive services initiatives and also to provide flexibility, scalability, high performance and reliability of the end-to-end system architecture.

Figure 34 illustrates the system components of a typical CMS platform.

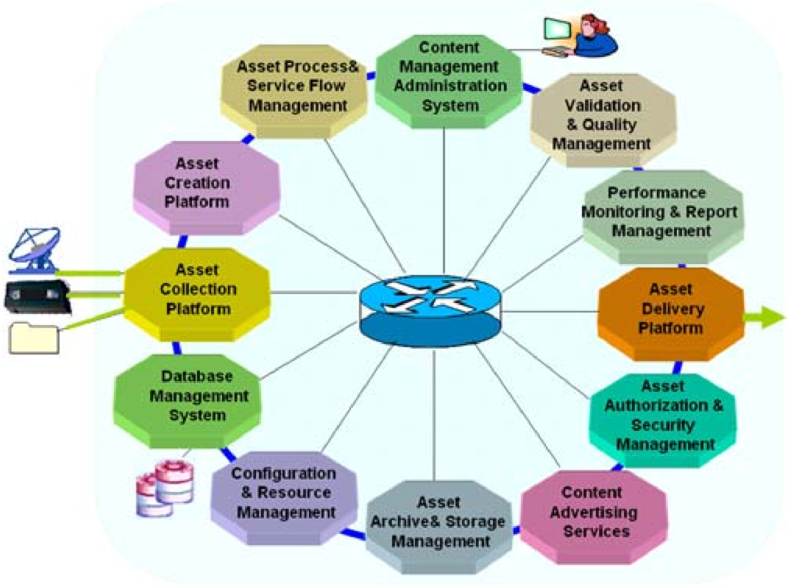

Asset Process& Service Flow Management
Content Management Administration System
Asset Validation & Quality Management
Asset Creation Platform
Performance Monitoring & Report Management
Asset Collection Platform
Asset Delivery Platform
Database Management System
Asset Authorization & Security Management
Configuration & Resource Management
Asset Archive& Storage Management
Content Advertising Services

**Figure 34: Content Management System**

The content processing services generally start with contractual agreements with the content providers. These agreements cover the way in which content is collected, validated, formatted, delivered, presented, reported and charged. The video content is ingested into the CMS such that it may be managed by the Asset Management functions including support for the major business functions of ingestion, preparation, programming, publishing, administration and reporting. The Content Management System Platform consists of the following subsystem components:

- CMS GUI, Administration, Operation and Control
- CMS Database Management System
- Asset Archive & Storage Management
- CMS Configuration & Resource Management

- Asset Process & Service Flow Management
- Asset Collection Platform
- Asset Validation & Quality Management
- Asset Authorization, Licensing & Security Management
- CMS Advertising Services
- Asset Delivery Platform
- CMS Performance Monitoring & Report Management
- Asset Creation Platform

### A.1.1 CMS Administration & Operation and Control

The CMS Administration and Operation Control System provides interfaces utilizing web browser and system console access and provides menus and system utilities to support database administration, system maintenance, system configuration, process control, scheduling, performance monitoring, reporting, license management, security and system configuration services. The Alarm & Alert function process notifies groups, individuals or other subsystems for action based on pre-defined criteria that indicate critical conditions, special conditions, warning, overdue conditions, hardware faults, discrepancy conditions, special return values, etc. This function is supported by all subsystems in the CMS end-to-end process flow.

CMS Administration & Operation function includes:

- Administration GUI using web client interface
- Administration access using the System Console
- Data Base Administration
- Storage allocation and de-allocation to subsystems
- Data & content backup & recovery
- Administration access control and account management
- System maintenance and network configuration
- Real-time system performance monitoring and content process status
- Historical system performance and activity reports
- Content collection system administration functions
- Content creation, modification, formatting and duplication process
- Metadata creation, modification, duplication and packaging
- Content validation, quality assurance and archiving
- Storage allocation and de-allocation configuration
- Asset distribution & scheduling
- Quality management and reports
- Subsystem reports, log files and recovery
- Alarms and alerts
- Contact administration

**Interface Requirements:**

TCP/IP, SNMP, Web, RS232 Console Interface

### A.1.2 CMS Database Management System

The CMS Management System supports all data storage, repository, replication, mirroring, access control and database functions required by the CMS subsystem. The data content may include: video, audio, metadata, package data, image, gaming executables, applications, messages, reports, logs, free format data and signaling information.

The CMS utilizes a global database management system distributed and replicated in SHE, CMC or data centers. This provides:

- Redundancy and disaster recovery for data storage and data recovery for the CMS platform
- Process data transfer between Commerce Engine, Advertising Management System and Report Management.

**Interface Requirements:**

TCP/IP

### A.1.3 Asset Archive & Storage Management

The Asset Archive & Storage Management subsystem provides the library management functions that is utilized to track the physical media received by the Asset Collection Subsystem. As part of the Asset Collection Platform functionalities, it receives physical media such as video tape, DVD data disks and external hard drives. These assets need to be bar coded and tracked as they move throughout the CMS during the workflow of content production. The Asset Archive & Storage Management subsystem utilizes the physical storage facilities for the depository of the physical assets. It also provides the storage management function that allocates storage, de-allocates storage and tracks and maintains storage locations for digital assets as they flow into and out of the CMS content delivery process. The actual system storage management of the CMS is supported by the CMS database Management System. The Asset Archive & Storage Management provides an additional layer of coordination to handle the quality assurance and resource management process for optimal storage usage.

**Interface Requirements:**

TCP/IP

### A.1.4 CMS Configuration & Resource Management

CMS Configuration & Resource Management subsystem, using the GUI interface, provides system utilities to manage:

- CMS subsystem software configuration
- CMS hardware configuration
- CMS network elements configuration
- Security configuration
- Hardware redundancy
- Subsystem process utilization and optimization

- Capacity management
- Network usage and bandwidth allocation
- Service and network quality assurance
- Originate service and network alerts

**Interface Requirements:**

TCP/IP, SNMP, Web, Console Interface

### A.1.5 Asset Process & Service Flow Management

Asset Process & Service Flow Management is the control mechanism that provides the collection, storage, quality control, staging, sequencing, reformatting, archiving, license management, security control, encryption, distributions and delivery process from entry into the CMS on through to the delivery of the processed content. It programmatically monitors the status of the work steps of each work item without requiring user intervention. A manual process might be defined and configured as part of process workflow. The CMS GUI & Administration System provides tools and system utilities to define the end-to-end process flow. It also provides the capability to update or duplicate the existing process flow to define a new one. The work flow process creation tools provides decision making steps based on the value of specific fields in the metadata file, content type, error code, content source, content type, content size, individual process steps in each subsystem, etc. It provides a real-time graphical presentation of the process flow status using the administration interface.

**Interface Requirements:**

TCP/IP

### A.1.6 Asset Collection Platform

The Asset Collection system provides interfaces, protocols, network connectivity, media interface, storage, servers and processing logic required to collect assets from content providers and deliver them to the CMS for further processing functions. The media entry points may include:

- Dockers & Catchers - Prepackaged VoD content delivery by satellite
- FTP - IP delivery from the content suppliers
- Physical Media - Including tapes, CDs, DVDs, hard drives and portable storage devices
- Web Application Servers – Including web media and B2B content delivery
- Live Stream Content

Once the content is successfully collected, the Asset Collection system instructs the Asset Archive & Storage subsystem to archive the content using the library management functions that later will be utilized to track the received physical media. These assets need to be bar coded and tracked as they will be addressed and identified by the process management function. The CSM provides support for live streaming of content directly delivered by content providers and received by the Asset Collection Platform and streamed to the Asset Delivery platform for further processing and possible content reformatting before delivery to the delivery servers.

The live stream content transport function provides content pass-through capabilities to the Asset Delivery system for further processing and possible reformatting of the content before delivery to the delivery servers.

**Interface Requirements:**

TCP/IP, UDP, HTTP, HTTPS, SOAP, XML

### A.1.7 Asset Validation & Quality Management

The Asset Validation & Quality Management system provides the content analysis, validation and quality assurance process to qualify the assets and certify them for delivery to Asset Delivery Platform. As part of the end-to-end process flow, assets that are collected by the Asset Collection Platform and assets modified and created locally by the Asset Creation Platform are required to be certified before release to delivery.

The Asset Validation & Quality Management system qualifies video content and metadata prior to delivery and assigns a quality grade approval. It also provides a description of rejection criteria.  If an asset is rejected it can be transferred to the Asset Creation Platform for correction automatically or manually utilizing the CMS GUI Administration interface.  It can also quantify the content based on end user viewing characteristics, bandwidth, CPE hardware and system environment.  In some cases, quality validation might require manual intervention.  The system provides capabilities in coordination with Asset Process and Service Flow management to define and control the handling and assignment of the asset in each stage.  As part of validation process the system  applies various automated techniques to determine if the content rating matches the define values.  Adult or offensive content not rated or user generated content is processed through the filtering rules pre-defined by the administration system.

### A.1.8 Asset Authorization, Licensing & Security Management

The Asset Authorization, Licensing & Security Management system provides coordination for content protection, usage authorization, content licensing and specific contractual usage conditions as agreed with content providers and mandated by service provider's business rules.  It also provides contract management, prescheduled license alert notification, content usage tracking and royalties.  Any release of a content package for production is certified by this system.  The contract mManagement and licensing data is stored in the CMS Database Management System and managed through the GUI and Administration System.

**Interface Requirements:**

TCP/IP

### A.1.9 CMS Advertising Services

As part of the advertising services process for banner ads, local linear broadcast advertising, target advertising, telescoping and VoD ad insertion services; ad contents, placement rules, configuration data and associated metadata files are requested by Campaign Manager to be created or changed by the CMS and delivered to advertising decision servers in VHO.  Also, in case of broadband and wireless services, similar processes are required to repackage and reformat ad content so that it is applicable to end users CPE capabilities.  The CMS Advertising Service Manager, as the agent of the advertising platform, coordinates the processing request and serves the signaling and communication process required to inform the Ad Decision Servers and deliver the ad content packages.

CSM Advertising Services can also coordinate the collection of ad scheduling data for linear broadcast programming services and target advertising and deliver to associated servers for distribution.

**Interface Requirements:**

TCP/IP, HTTP, HTTPS, XML, SOAP

### A.1.10 Asset Delivery Platform

The Asset Delivery Platform provides interfaces, protocols, network connectivity, transcoding, reformatting and processing logic required to deliver assets to the Asset Management systems in the SHE and VHO and also to CDN platforms for broadband and wireless services. Once the content is successfully collected, certified, licensed and released it is assigned to the Asset Delivery Platform for delivery and ingestion processing. In case of non-TV services, content might require real-time streaming, reformatting and transcoding based on end user's viewing characteristics, bandwidth availability, CPE hardware and system environment. Real-time formatting, resizing and transcoding functions are carefully evaluated where preprocessing of this function by the Asset Creation Platform might be more efficient. Comparison of the expense and effort associated with storing different preprocessed content formats rather than storing a fixed content format and then using a real-time reformatting process determines the right approach.

The Asset Delivery Platform provides capabilities for live streaming of content directly delivered by content providers and received by the Asset Collection Platform and streamed to the Asset Delivery platform for further processing and possible reformatting of the content before delivery to the CDN.

The Asset Delivery Platform provides content creation, conversion, reformatting, resizing and transcoding in the following formats in an offline processing environment:

SD, HD, MPEG2, H.264, AVI, QuickTime, Windows Media, Flash, Real,

The Asset Delivery Platform provides conversion, reformatting, resizing and transcoding in the following formats in a real-time streaming process:

H.264, AVI, QuickTime, Windows Media, Flash, Real

The Asset Delivery Platform provides real-time streaming in the following formats from the preformatted local content or from the received video streams from an external video content provider:

SD, HD, MPEG2, H.264, AVI, QuickTime, Windows Media, Flash, Real

**Interface Requirements:**

TCP/IP, UDP/IP, RTSP, SIP, FTP, HTTP, HTTPS, SOAP, XML, MPG2, H.264

### A.1.11 CMS Performance Monitoring & Report Management

The Performance Monitoring & Report Management system provides real-time monitoring of content processed through the CMS and managed by the Process & Service Flow Management System. At each stage of task completion, as defined by process flow, each subsystem unit reports completion status and failure causes to reporting system. This data is dynamically processed and presented graphically in a dashboard interface by the GUI Administration system. The Reporting Engine collects reports across all system units, aggregates the data and stores them in the database. The report management system provides various standard reports based on collection period and content details. It shall also provides the mechanism and system utilities to define ad-hoc reports dynamically.

**Interface Requirements:**

TCP/IP, HTTP, HTTPS, XML, SOAP, FTP

### A.1.12 Asset Creation Platform

The Asset Creation Platform is an integration of automated video processing components complemented with various manual processing utilities that provides capabilities for content creation, content edition, content enhancements, quality review, validation, rating, decryption, encryption, transcoding, reformatting, trouble shooting, branding, duplication, ad insertion and resizing of assets locally. The assets include video content, ad content, audio, music, metadata, package data, image, gaming executables, applications, messages, reports, logs, free format data and signaling information. Where required and authorized, the Asset Creation Platform provides capabilities to decrypt content to process, reformat, encrypt, DRM wrapping and repackaging for release and distribution processing. Packages that are released are archived for long-term production usage. The Asset Creation Platform manages metadata as an asset component and provides system function and utilities to import, change, create and publish metadata as an asset component. The same applies to images, audio, graphics and text assets. Creation and enhancement of metadata capabilities are critical components for production of ad content and associated metadata files. The Asset Creation Platform processes and packages the VoD content and associated data and metadata files and submits them to an archiving subsystem for storing in a database library and for release and distribution to the Asset Distribution Platform. As the VoD assets are packaged and released for production, the Catalog Library is updated with content classification and communicated to the Interactive Program Guide (IPG) for distribution to the end user and Catalog Management Platform residing in the CDN platform.

As part of the advertising services process for banner ads and targeted advertising, ad contents, placement rules and associated metadata files are requested by the Campaign Manager and communicated to the Asset Creation Platform by the Advertising Service Manager.

The Asset Delivery Platform provides conversion, reformatting, resizing and transcoding in the following formats in a real-time streaming process:

H.264, AVI, QuickTime, Windows Media, Flash, Real

The Asset Delivery Platform provides real-time streaming in the following formats from preformatted local content or by receiving video streams from an external video content provider:

SD, HD, MPEG2, H.264, AVI, QuickTime, Windows Media, Flash, Real

**Interface Requirements:**

TCP/IP, HTTP, HTTPS, XML, SOAP, FTP

## *A.2 Content Management System Design & Integration*

The end-to-end content processing work flow, from entry into the system until downloaded or streamed to the assigned destination target, is pre-planned and managed step-by-step utilizing a deterministic state machine approach. Contracts between service provider and content providers define requirements for some of the rules and process functions that are needed and utilized in the end-to-end process flow and are specific for each content class. Content class is defined as the group of content units that share a common treatment from beginning to end. The contract rules are defined and managed by the contract management system and stored in the database utilizing the CMS administration system GUI interface. Contract specific functions related to licensing, encryption, key management, DRM, protection, security and royalties are handled by the Asset Authorization and Security Management subsystem.

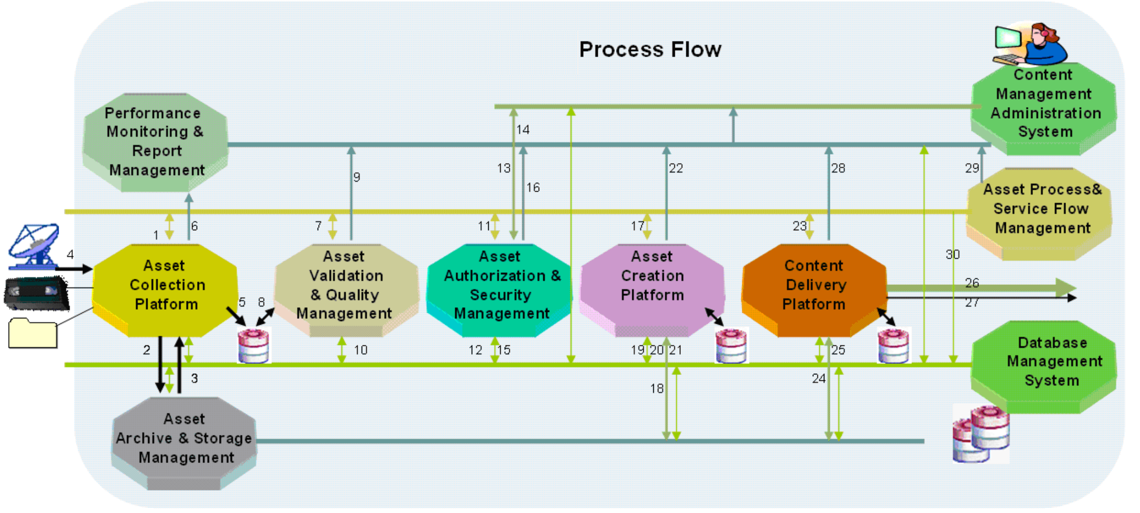

Process Flow
Content Management Administration System
Performance Monitoring & Report Management
Asset Process& Service Flow Management
Asset Collection Platform
Asset Validation & Quality Management
Asset Authorization & Security Management
Asset Creation Platform
Content Delivery Platform
Database Management System
Asset Archive & Storage Management
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

**Figure 35: Content Management System Process Work Flow**

Process work flow for each content class is created using the interactive system and graphical utilities provided by the Asset Process & Service Flow Management and are stored in the database. This subsystem communicates to all other CMS system units and manages decision making steps based on various pre-defined criteria including error codes, original content format, target content format, schedules, value of specific fields in the metadata file, contract rules, collecting protocols, delivery protocols, etc.

To clarify how the end to end process flow works, the following process describes a typical scenario with no error conditions. The purpose of this scenario is for better understanding of the design concept.

1. Asset Process & Service Flow Management assigns a work unit to the Asset Collection Platform. A work unit is a set of system tasks that are needed to complete a segment of the content processing function.
2. The Asset Collection Platform communicates to the Asset Archive & Storage Management system and requests resources and data needed to complete the work unit.
3. The Asset Archive & Storage Management system evaluates the request and, in communication with the Database Management system, allocates resources and makes them available to the Asset Collection Platform.

4. The Asset Collection Platform initializes the resources and communicates to the assigned media to receive the content.
5. The Asset Collection Platform collects the asset required and stores it in the allocated storage location.
6. The Asset Collection Platform reports to the Performance Monitoring and Report Management system the status and completion of the collection process.  It also reports to the Asset Process & Service Flow Management system the completion of the work unit and description of the work that was completed and any error conditions.
7. The Asset Process & Service Flow Management system evaluates the asset collection results and based on no error conditions, assigns Asset Validation and Quality Management to review the asset, validate it against any errors or inconsistencies and execute the quality assurance process.
8. Asset Validation and Quality Management receives the work unit assignment and allocates the storage and processes the asset.  It executes the validation and service assurance work.
9. Asset Validation and Quality Management reports to the Performance Monitoring and Report Management system the status and the completion of the validation process.  It also report to the Asset Process & Service Flow Management system the completion of the work unit and description of the work that was completed and if any error conditions.
10. Asset Validation and Quality Management updates the database with the validation results and release of the asset.
11. Asset Process & Service Flow Management evaluates the asset validation results and, based on no error conditions, assigns Asset Authorization and Security Management to process the licensing, issue encryption key (if required) and certification process.
12. The Asset Authorization and Security Management will retrieve the contractual authorization and licensing data from the database and issues the encryption key (if required) and execute certification process.
13. In case a manual intervention is required by the CMS Operation staff to get approval in certain conditions, a message for approval request is sent to the CMS Administration system.
14. Upon approval of the licensing authorization and receiving the confirmation, Authorization and Security Management finalizes the certification process and releases the asset.
15. The Authorization and Security Management system updates the database with the certification and asset release data.
16. Authorization and Security Management reports to the Performance Monitoring and Report Management system the status and completion of the certification and release process.  It also reports to the Asset Process & Service Flow Management system the completion of the work unit and description of the work that was completed and any error conditions.
17. Asset Process & Service Flow Management evaluates the asset authorization results and, based on no error conditions, assigns the work unit to the Asset Creation Platform to execute the reformatting, encryption, resizing, transcoding and content enhancement work as required.
18. The Asset Creation Platform receives the work unit assignment.  It communicates to Asset Archive and Storage Management to allocate the storage and resources that are required to execute the process.
19. The Asset Creation Platform retrieves the content, the data files and related information required to execute the work, from the database.
20. The Asset Creation Platform retrieves the content, the data files and related information required to execute the work from the database.

21. The Asset Creation Platform executes the work unit and updates the database with the enhanced content.
22. The Asset Creation Platform reports to the Performance Monitoring and Report Management system the status and completion of the content enhancement work. It also reports to the Asset Process & Service Flow Management system the completion of the work unit and description of the work that was completed and any error conditions.
23. Asset Process & Service Flow Management evaluates the asset creation results and, based on no error conditions, assigns the work unit to the Asset Delivery Platform to execute the delivery process to the target system assigned for this asset.
24. The Asset Delivery Platform allocates storage, resource, content and network bandwidth required to execute the work.
25. The Asset Delivery Platform retrieves the data files, configuration, network protocols and related information required to execute the delivery process. The delivery process can be content download, streaming or file transfer to one or multiple destinations as instructed by the work unit.
26. The Asset Delivery Platform executes the content delivery process to the destinations assigned to the work unit
27. The Asset Delivery Platform may also, in addition to video content, deliver package data, metadata or related information to the same system destination or alternative locations.
28. The Asset Delivery Platform reports to the Performance Monitoring and Report Management system the status and completion of the content delivery work. It also reports to the Asset Process & Service Flow Management system the completion of the work unit and description of the work that was completed and any error conditions.

## APPENDIX B OPENSTREAM SERVICES ARCHITECTURE (INFORMATIVE)

The *OpenStream Services Architecture* (OSA) specification defines interfaces between various components in legacy deployed video on demand systems.

The *OpenStream Services Architecture* specification may be found at:

> http://www.ericsson.com/res/thecompany/docs/journal_conference_papers/service_layer/osa10_ericsson_public.pdf

The OSA specification defines many interfaces, some of which are within the scope of ATIS-IIF CoD Service specification. Specifically, the Content, Session and Stream components that are described in sections 8, 12, and 15 of the OSA specification are relevant to messages passed between the IPTV Control Functions and the Content Distribution & Delivery Functions. These functionally map to the C5, S1 and S5 reference points in ATIS-0800042.

Figure 36 illustrates the correlation between the OpenStream and the ATIS-0800042 reference architectures.

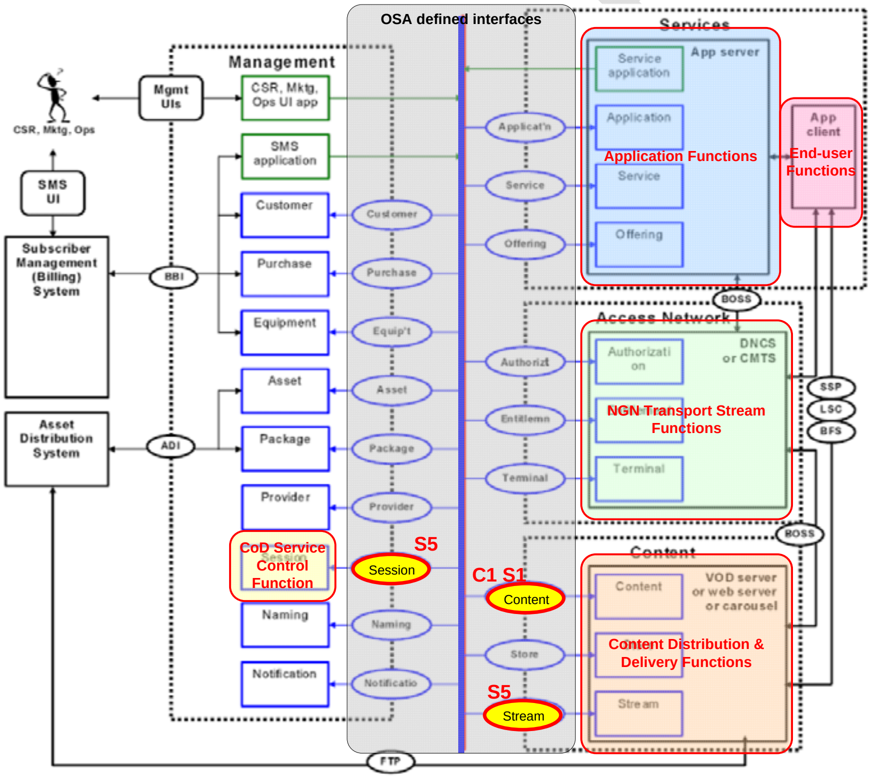


**Figure 36: Correlation between OSA and ATIS IIF CoD**

Figure 37 corresponds to Figure 7 in ATIS-0800042 and shows the OSA method for signaling the Content Origin Function that an asset is available.  The Content::provision() message contains the URI of the content that the Content Origin Function uses to retrieve the content.

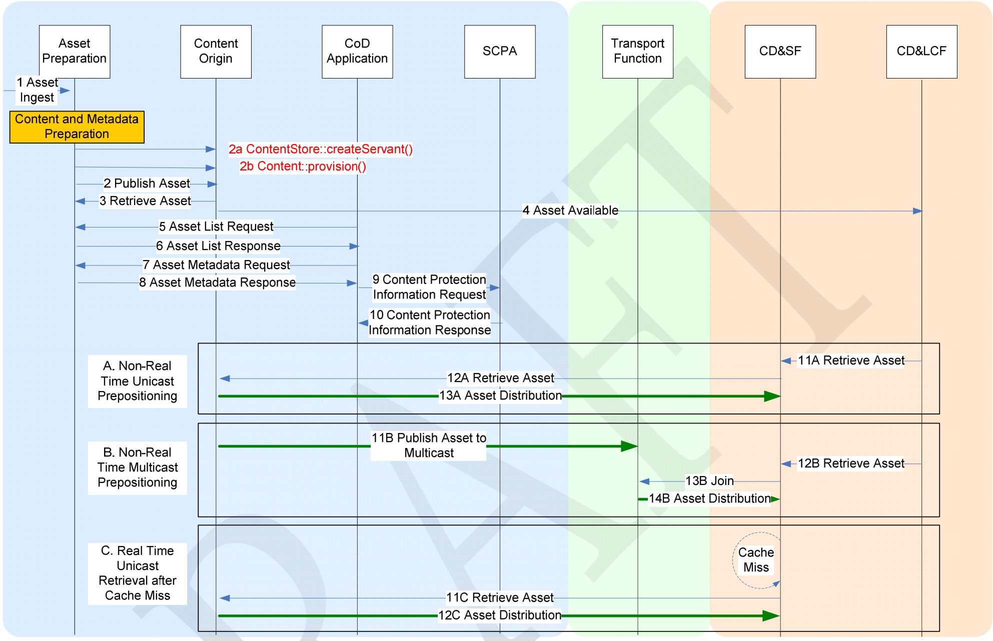


**Figure 37: Content Preparation and Distribution (OSA Method)**

Figure 38 corresponds to Figure 10 in ATIS-0800042 and shows the OSA method for locating the CD&SF in steps 5 and 6 and subsequently informing the CD&SF of the content that is being requested in step 7. Step 7a is an optional step that allows the CD&SF to request resources. If this message is used, the response is sent back in step 10a.

**Deleted:** Figure 10

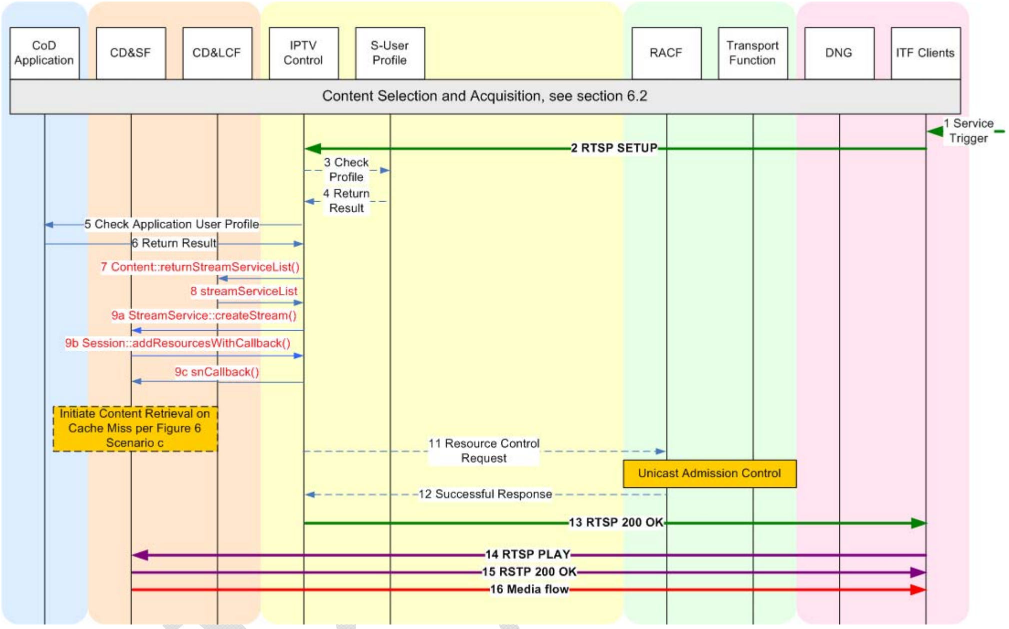

CoD Application
CD&SF
CD&LCF
IPTV Control
S-User Profile
RACF
Transport Function
DNG
ITF Clients
Content Selection and Acquisition, see section 6.2
1 Service Trigger
2 RTSP SETUP
3 Check Profile
4 Return Result
5 Check Application User Profile
6 Return Result
7 Content::returnStreamServiceList()
8 streamServiceList
9a StreamService::createStream()
9b Session::addResourcesWithCallback()
9c snCallback()
Initiate Content Retrieval on Cache Miss per Figure 6 Scenario c
11 Resource Control Request
Unicast Admission Control
12 Successful Response
13 RTSP 200 OK
14 RTSP PLAY
15 RSTP 200 OK
16 Media flow

**Figure 38: Non-IMS Proxy CoD Session Establishment – RTSP (OSA Method)**

Figure 39 corresponds to Figure 12 in ATIS-0800042 and shows the OSA method of session termination that is initiated by the ITF.

**Deleted:** Figure 12

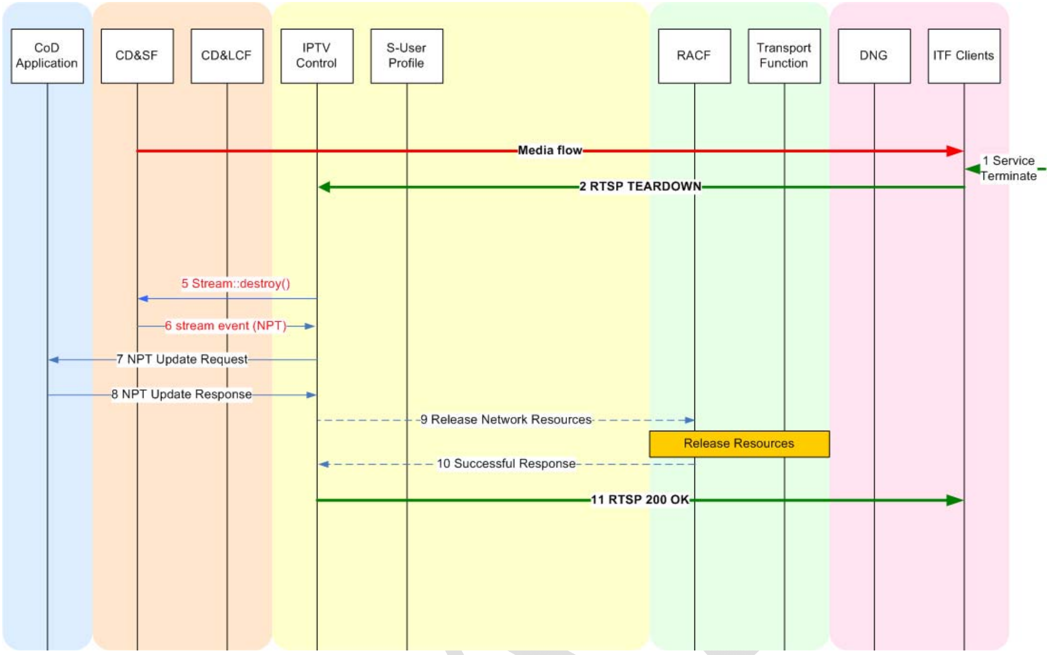


**Figure 39: Non-IMS Proxy CoD Session Termination- RTSP (OSA Method)**

Figure 40 corresponds to Figure 22 in ATIS-0800042 and shows the OSA method for session establishment in the IMS case.  Steps 13, 14, and 15 show the messages for locating the CD&SF and subsequently informing the CD&SF of the content that is being requested.  There is no opportunity for the CD&SF to request resources as shown in Figure 38 for the non-IMS case.

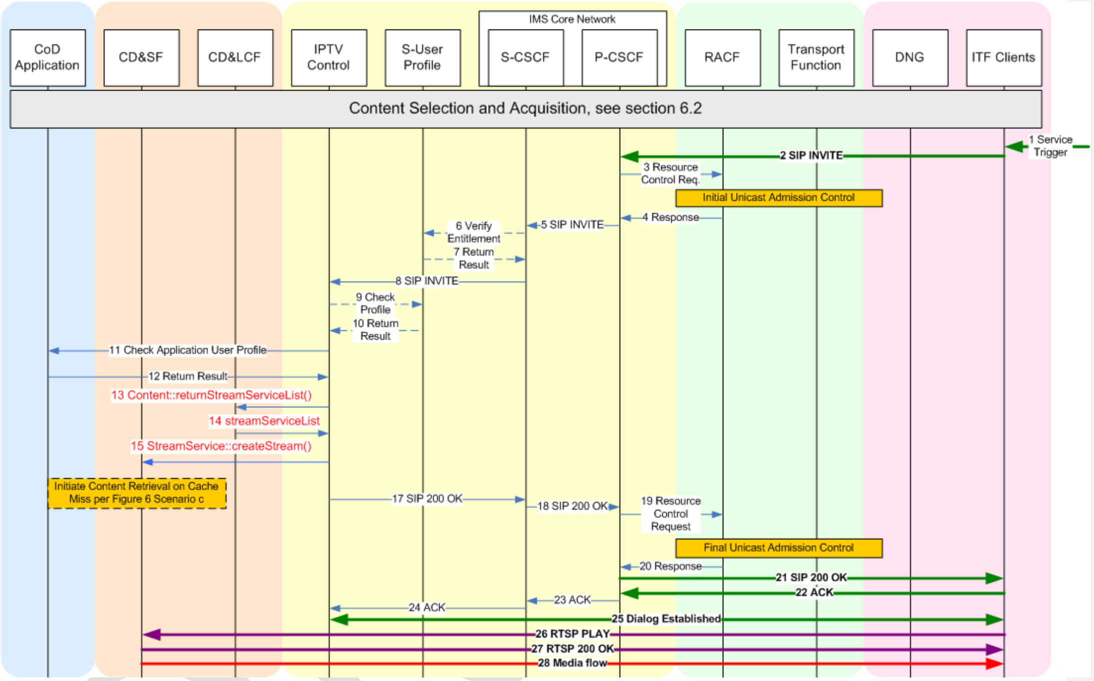


**Figure 40: IMS CoD Session Establishment – RTSP (OSA Method)**

Figure 41 corresponds to Figure 15 in ATIS-0800042  and shows the OSA method of session termination initiated by the ITF in the IMS case.

**Deleted:** Figure 15

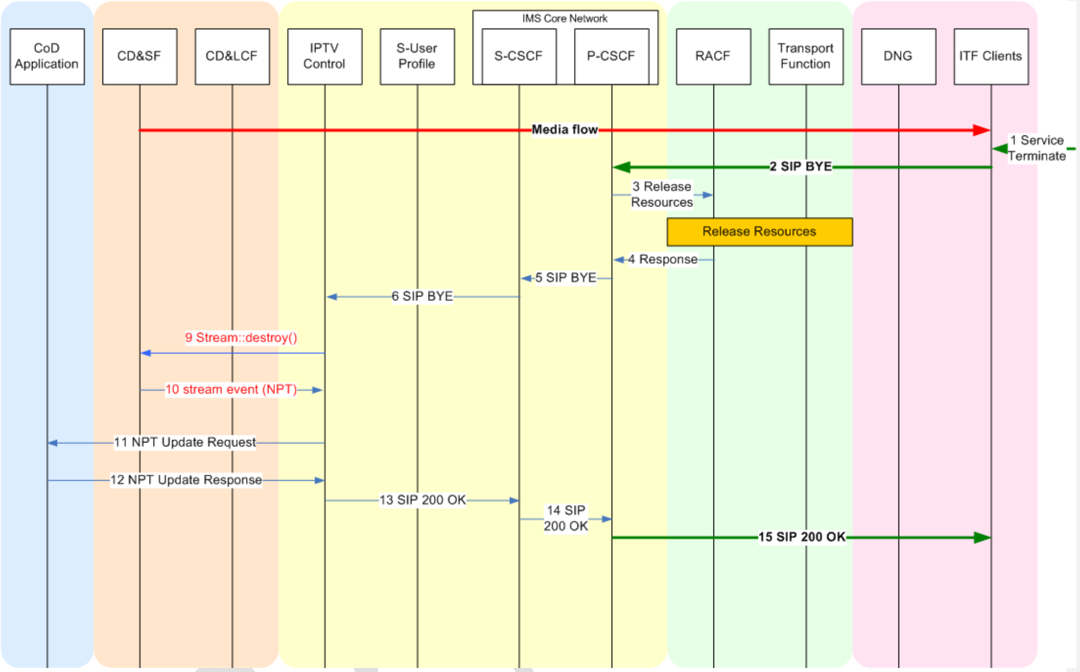


**Figure 41: IMS CoD Session Termination – RTSP (OSA Method)**