

ONVIF™

# ONVIF Media Test Specification

Version 1.02.4

July, 2011

© 2011 by ONVIF, Inc. All rights reserved.

Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

### Revision History

Ver.	Date	Description
1.02.4	29th/Jul, 2011	First issue of Media Test Specification

## Table of Contents

1	Introduction .....	10
1.1	Scope .....	10
1.1.1	Media Configuration .....	10
1.1.2	Real Time Streaming .....	11
2	Terms and Definitions .....	12
2.1	Definitions.....	12
2.2	Abbreviations.....	12
3	Test Overview.....	13
3.1	Test Setup .....	13
3.1.1	Network Configuration for DUT .....	13
3.2	Prerequisites .....	14
3.3	Requirement Level .....	14
3.4	Test Policy .....	14
3.4.1	Media Configuration .....	14
3.4.2	Real Time Streaming .....	15
4	Media Configuration Test Cases .....	16
4.1	Media Profile .....	16
4.1.1	MEDIA PROFILE CONFIGURATION.....	16
4.1.2	DYNAMIC MEDIA PROFILE CONFIGURATION .....	17
4.1.3	PROFILES CONSISTENCY .....	21
4.2	Video Configuration.....	23
4.2.1	VIDEO SOURCE CONFIGURATION .....	23
4.2.2	VIDEO ENCODER CONFIGURATION .....	27
4.2.3	JPEG VIDEO ENCODER CONFIGURATION.....	29
4.2.4	MPEG4 VIDEO ENCODER CONFIGURATION.....	32
4.2.5	H.264 VIDEO ENCODER CONFIGURATION.....	35
4.2.6	GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES .....	38
4.3	Video Source Configuration .....	40
4.3.1	VIDEO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY .....	40
4.3.2	VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION CONSISTENCY .....	41
4.3.3	VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION OPTIONS	

CONSISTENCY .....	43
4.3.4     PROFILES AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY .....	45
4.3.5     VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCES CONSISTENCY .....	47
4.3.6     VIDEO SOURCE CONFIGURATION USE COUNT (CURRENT STATE).....	49
4.3.7     VIDEO SOURCE CONFIGURATION USE COUNT (ADD SAME VIDEO SOURCE CONFIGURATION TO PROFILE TWICE) .....	51
4.3.8     VIDEO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO SOURCE CONFIGURATIONS IN PROFILE) .....	55
4.3.9     VIDEO SOURCE CONFIGURATION USE COUNT (REMOVE VIDEO SOURCE CONFIGURATION) .....	60
4.3.10    VIDEO SOURCE CONFIGURATION USE COUNT (DELETION PROFILE WITH VIDEO SOURCE CONFIGURATION) .....	63
4.3.11    VIDEO SOURCE CONFIGURATION USE COUNT (SET VIDEO SOURCE CONFIGURATION) .....	66
4.4     Video Encoder Configuration .....	68
4.4.1     VIDEO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY .....	68
4.4.2     VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION CONSISTENCY .....	70
4.4.3     VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY .....	72
4.4.4     PROFILES AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY .....	74
4.4.5     VIDEO ENCODER CONFIGURATION USE COUNT (CURRENT STATE).....	76
4.4.6     VIDEO ENCODER CONFIGURATION USE COUNT (ADD SAME VIDEO ENCODER CONFIGURATION TO PROFILE TWICE) .....	78
4.4.7     VIDEO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO ENCODER CONFIGURATIONS IN PROFILE) .....	83
4.4.8     VIDEO ENCODER CONFIGURATION USE COUNT (REMOVE VIDEO ENCODER CONFIGURATION) .....	87
4.4.9     VIDEO ENCODER CONFIGURATION USE COUNT (PROFILE DELETION WITH VIDEO ENCODER CONFIGURATION) .....	90
4.4.10    VIDEO ENCODER CONFIGURATION USE COUNT (SET VIDEO ENCODER CONFIGURATION) .....	94
4.5     Audio Configuration .....	96

4.5.1	AUDIO SOURCE CONFIGURATION.....	96
4.5.2	AUDIO ENCODER CONFIGURATION .....	100
4.5.3	G.711 AUDIO ENCODER CONFIGURATION .....	105
4.5.4	G.726 AUDIO ENCODER CONFIGURATION.....	108
4.5.5	AAC AUDIO ENCODER CONFIGURATION .....	111
4.5.6	GET AUDIO SOURCE CONFIGURATION – INVALID CONFIGURATIONTOKEN .....	114
4.5.7	GET AUDIO SOURCE CONFIGURATION OPTIONS .....	115
4.5.8	GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID PROFILETOKEN.....	118
4.5.9	GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID CONFIGURATION TOKEN 119	
4.5.10	SET AUDIO SOURCE CONFIGURATION – INVALID TOKEN .....	120
4.6	Audio Source Configuration .....	121
4.6.1	AUDIO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY .....	121
4.6.2	AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION CONSISTENCY .....	123
4.6.3	AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION OPTIONS CONSISTENCY .....	125
4.6.4	PROFILES AND AUDIO SOURCE CONFIGURATION OPTIONS CONSISTENCY .....	127
4.6.5	AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCES CONSISTENCY.....	129
4.6.6	AUDIO SOURCE CONFIGURATION USE COUNT (CURRENT STATE) .....	131
4.6.7	AUDIO SOURCE CONFIGURATION USE COUNT (ADD SAME AUDIO SOURCE CONFIGURATION TO PROFILE TWICE).....	133
4.6.8	AUDIO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO SOURCE CONFIGURATIONS IN PROFILE) .....	137
4.6.9	AUDIO SOURCE CONFIGURATION USE COUNT (REMOVE AUDIO SOURCE CONFIGURATION).....	142
4.6.10	AUDIO SOURCE CONFIGURATION USE COUNT (PROFILE DELETION WITH AUDIO SOURCE CONFIGURATION) .....	145
4.6.11	AUDIO SOURCE CONFIGURATION USE COUNT (SET AUDIO SOURCE CONFIGURATION).....	148
4.7	Audio Encoder Configuration .....	150
4.7.1	AUDIO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY .....	150
4.7.2	AUDIO ENCODER CONFIGURATIONS AND AUDIO ENCODER CONFIGURATION	

CONSISTENCY .....	152
4.7.3 AUDIO ENCODER CONFIGURATIONS AND AUDIO ENCODER CONFIGURATION OPTIONS CONSISTENCY .....	154
4.7.4 PROFILES AND AUDIO ENCODER CONFIGURATION OPTIONS CONSISTENCY .....	156
4.7.5 AUDIO ENCODER CONFIGURATION USE COUNT (CURRENT STATE) .....	158
4.7.6 AUDIO ENCODER CONFIGURATION USE COUNT (ADD SAME AUDIO ENCODER CONFIGURATION TO PROFILE TWICE) .....	160
4.7.7 AUDIO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO ENCODER CONFIGURATIONS IN PROFILE) .....	165
4.7.8 AUDIO ENCODER CONFIGURATION USE COUNT (REMOVE AUDIO ENCODER CONFIGURATION) .....	169
4.7.9 AUDIO ENCODER CONFIGURATION USE COUNT (DELETION PROFILE WITH AUDIO ENCODER CONFIGURATION) .....	172
4.7.10 AUDIO ENCODER CONFIGURATION USE COUNT (SET AUDIO ENCODER CONFIGURATION) .....	176
4.8 PTZ Configuration .....	178
4.8.1 PTZ CONFIGURATION .....	178
4.8.2 PTZ CONFIGURATIONS AND PROFILES CONSISTENCY .....	180
4.9 Metadata Configuration .....	182
4.9.1 METADATA CONFIGURATION .....	182
4.10 Media Streaming .....	186
4.10.1 SNAPSHOT URI .....	186
4.11 Error Handling .....	188
4.11.1 SOAP FAULT MESSAGE .....	188
4.11.2 SOAP FAULT MESSAGE .....	189
4.11.3 START MULTICAST - INVALID PROFILE TOKEN .....	191
5 Real Time Streaming Test Cases .....	192
5.1 Video Streaming .....	192
5.1.1 MEDIA CONTROL – RTSP/TCP .....	192
5.1.2 MEDIA STREAMING – RTSP KEEPALIVE (SET_PARAMETER) .....	196
5.1.3 MEDIA STREAMING - RTSP KEEPALIVE (OPTIONS) .....	200
5.1.4 MEDIA STREAMING – JPEG (RTP-Unicast / UDP) .....	203
5.1.5 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP) .....	207

5.1.6	MEDIA STREAMING – JPEG (RTP/RTSP/TCP) .....	212
5.1.7	MEDIA STREAMING – MPEG4 (RTP-Unicast/ UDP) .....	215
5.1.8	MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP) .....	219
5.1.9	MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP) .....	224
5.1.10	SET SYNCHRONIZATION POINT – MPEG4 .....	227
5.1.11	MEDIA STREAMING – H.264 (RTP-Unicast/ UDP) .....	232
5.1.12	MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP) .....	235
5.1.13	MEDIA STREAMING – H.264 (RTP/RTSP/TCP) .....	240
5.1.14	SET SYNCHRONIZATION POINT – H.264 .....	244
5.1.15	MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4) .....	248
5.1.16	MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4) .....	251
5.1.17	MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4) .....	255
5.2	Audio Streaming .....	259
5.2.1	MEDIA STREAMING – G.711 (RTP-Unicast/UDP) .....	259
5.2.2	MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP) .....	263
5.2.3	MEDIA STREAMING – G.711 (RTP/RTSP/TCP) .....	267
5.2.4	MEDIA STREAMING – G.726 (RTP-Unicast/UDP) .....	271
5.2.5	MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP) .....	275
5.2.6	MEDIA STREAMING – G.726 (RTP/RTSP/TCP) .....	279
5.2.7	MEDIA STREAMING – AAC (RTP-Unicast/UDP) .....	283
5.2.8	MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) .....	287
5.2.9	MEDIA STREAMING – AAC (RTP/RTSP/TCP) .....	291
5.3	Audio & Video Streaming .....	296
5.3.1	MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/ UDP) .....	296
5.3.2	MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP) .....	300
5.3.3	MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP) .....	304
5.3.4	MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/ UDP) .....	308
5.3.5	MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP) .....	312
5.3.6	MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP) .....	316
5.3.7	MEDIA STREAMING – JPEG/AAC (RTP-Unicast/ UDP) .....	320
5.3.8	MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP) .....	324
5.3.9	MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP) .....	328
5.4	Notification Streaming Interface .....	332



5.4.1	NOTIFICATION STREAMING .....	332
6	Annex .....	337
A.1	Invalid Media Profile .....	338
A.2	Invalid RTP Header .....	338
A.3	StreamSetup syntax for GetStreamUri .....	338
A.4	I-frame insertion time interval .....	339
A.5	Media Profile Configuration for Video Streaming .....	340
A.6	Media Profile Configuration for Audio & Video Streaming.....	341
A.7	MetadataConfiguration for receiving / not receiving events metadata .....	345
A.8	Multicast specific field in SDP (RTSP DESCRIBE response) .....	345

## 1 Introduction

The goal of the ONVIF test specification set is to **make it possible to realize** fully interoperable IP physical security implementations from different **vendors**. The set of ONVIF test specification describes the test cases needed to verify the [ONVIF Core] requirements. **It also describes the test framework, test setup, pre-requisites, test policies needed for the execution of the described test cases.**

This ONVIF Media Test Specification acts as a **supplementary** document to the [ONVIF Core], clarifying the **requirements** wherever needed. And also this specification acts an **input document** to the development of test tool which will be used to test the test the ONVIF device implementation conformance towards the [ONVIF Core]. **This test tool is referred as Network Video Client (NVC) hereafter.**

### 1.1 Scope

This ONVIF Media Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant devices. Conformance testing is meant to be functional black-box testing. The objective of this specification to provide test cases to test individual requirements of ONVIF devices according to ONVIF Media Service and the Realtime Streaming Specification which is defined in [ONVIF Core].

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide **comprehensive** test suite coverage for [ONVIF Core].
- This specification does not address the following.

Product use cases and non-functional (performance and regression) testing.

- SOAP Implementation Interoperability test i.e. Web Service Interoperability Basic Profile version 2.0 (WS-I BP 2.0).
- Network protocol implementation Conformance test for HTTP, HTTPS, RTP and RTSP protocol.
- **Poor streaming performance test** (audio/video distortions, missing audio/video frames, incorrect lib synchronization etc.).
- Wi-Fi Conformance test

**The set of ONVIF Test Specification will not cover the complete set of requirements as defined in [ONVIF Core];** instead it would cover subset of it. The scope of this specification is to derive all the normative requirements of [ONVIF Core] which is related to ONVIF Media Service and Realtime Streaming Specification and some of the optional requirements.

This ONVIF Media Test Specification covers ONVIF Media service and Real-time Streaming specification which is a functional block of [ONVIF Core]. The following sections describe the brief overview of and scope of each functional block.

#### 1.1.1 Media Configuration

Media Configuration section covers the test cases needed for the verification of media service features as mentioned in [ONVIF Core]. Media service is used to configure the media and other real time streaming configurations.

Briefly it covers the following things

- Manage media profiles.
- Manage configuration entities.
- Initiate and manipulate Audio/Video streams for different media formats.

The scope of this specification is to cover following configuration entities and Audio/Video media formats.

- Configuration Entities:
- Video source configuration
- Audio source configuration
- Video encoder configuration
- Audio encoder configuration
- PTZ configuration
- Metadata configuration

Video Codec:

- JPEG QVGA
- MPEG-4, Simple Profile
- H.264, Baseline

Audio Codec:

- G.711
- G.726
- AAC

### 1.1.2 Real Time Streaming

Real Time Streaming covers the test cases needed for the verification of Real time streaming features as mentioned in [ONVIF Core]. Real time streaming section defines different media streaming options based on RTP for video, audio and metadata streams. Media control is done using RTSP protocol.

The scope of this specification to cover the following real time streaming options for JPEG, MPEG4 and H.264 video streams, and JPEG/ G.711, JPEG/ G.726 and JPEG/ AAC Audio & Video streams.

- RTSP control requests
- RTP unicast over UDP
- RTP over RTSP over TCP
- RTP over RTSP over HTTP over TCP
- RTCP

## 2 Terms and Definitions

### 2.1 Definitions

This section defines terms that are specific to the ONVIF Media Service and tests. For a list of applicable general terms and definitions, please see [ONVIF Base Test].

<b>Configuration Entity</b>	A network video device media abstract component that is used to produce a media stream on the network, i.e. video and/or audio stream.
<b>Media Profile</b>	A media profile <b>maps</b> a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations.

### 2.2 Abbreviations

This section describes abbreviations used in this document.

AAC	Advanced Audio Coding
JPEG	Joint Photographic Experts Group
MPEG-4	Moving Pictures Experts Group-4
QVGA	Quarter Video Graphics Array
TTL	Time To Live

### 3 Test Overview

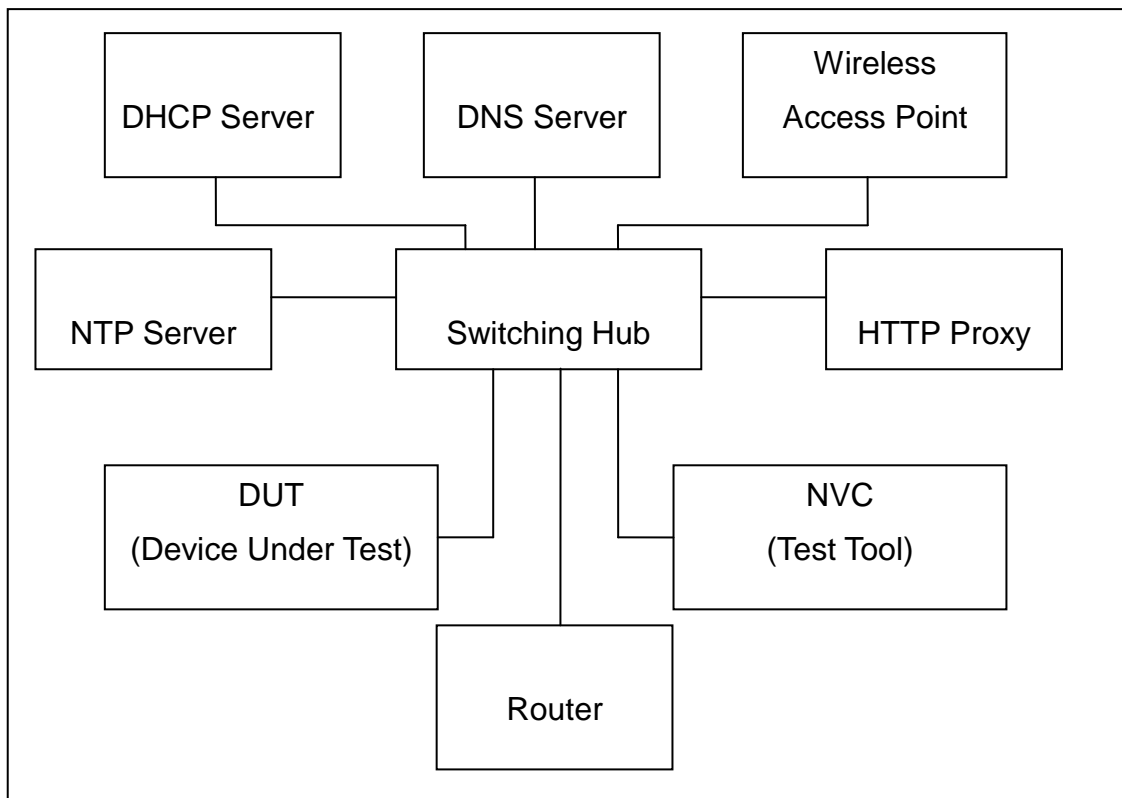
This section describes about the test setup and prerequisites needed, and the test policies that should be followed for test case execution.

#### 3.1 Test Setup

##### 3.1.1 Network Configuration for DUT

The generic test configuration for the execution of test cases defined in this document is as shown below (Figure 1)

Based on the individual test case requirements, some of the entities in the below setup **may not be needed** for the execution of those corresponding test cases.



**Figure 1: Test Configuration for DUT**

**DUT:** ONVIF device to be tested. Hereafter, this is referred to as DUT (Device Under Test).

**NVC Test Tool:** Tests are executed by this system and it controls the behaviour of the DUT. It handles both expected and unexpected behaviour.

**HTTP Proxy:** provides facilitation in case of RTP and RTSP tunneling over HTTP.

**Wireless Access Point:** provides wireless connectivity to the devices that support wireless connection.

**DNS Server:** provides DNS related information to the connected devices.



**DHCP Server:** provides IPv4 Address to the connected devices.

**NTP Server:** provides time synchronization between NVC and DUT.

Switching Hub: provides network connectivities among all the test equipments in the test environment. All devices should be connected to the Switching Hub.

Router: provides router advertisements for IPv6 configuration.

### 3.2 Prerequisites

The pre-requisites for executing the test cases described in this Test Specification are

- The DUT must be configured with an IPv4 address.
- The DUT must be IP reachable [in the test configuration].
- The DUT must be able to be discovered by the NVC Test Tool.
- The DUT must be configured with the time i.e. manual configuration of UTC time and if NTP is supported by DUT then NTP time must be synchronized with NTP Server.
- The DUT time and NVC Test tool time must be synchronized with each other either manually or by common NTP server.

### 3.3 Requirement Level

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119]. Additionally, this specification describes the key words "MUST IF SUPPORTED", "SHOULD IF SUPPORTED", "MUST IF IMPLEMENTED [A]", "MUST IF SUPPORTED [A] & IMPLEMENTED [B]", "SHOULD IF IMPLEMENTED [A]", "SHOULD IF SUPPORTED [A] & IMPLEMENTED [B]". For the details on how the requirement levels affect the test cases described in this specification, refer to [ONVIF Test].

### 3.4 Test Policy

This section describes the test policies specific to the test case execution of each functional block.

The DUT must adhere to the test policies defined in this section.

#### 3.4.1 Media Configuration

Prior to the execution of Media Configuration test cases, DUT must be discovered by NVC and it must demonstrate media capabilities to NVC using device management service.

**DUT must support at least one media profile with Video Configuration. Video Configuration must include video source and video encoder media entities.**

**DUT must support JPEG QVGA encoding.**

NVC user must **explicitly** specify the optional media formats supported by DUT.

NVC user must explicitly specify if the DUT supports **Audio and PTZ**.

**DUT must allow for creation of at least one media profile by NVC.** In certain test cases, NVC may create new media configuration (i.e. media profile and media entities). In such cases, the test procedure will delete those modified configuration at the end of the test procedure.

DUT should respond with proper response message for all SOAP actions. Sending fault messages such as "ter:ConfigurationConflict" will be treated as FAILURE of the test cases.

Please refer to Section 4 for Media Configuration Test Cases.

### 3.4.2 Real Time Streaming

Real time streaming test case execution would need the successful execution of some of the Media Configuration test cases. So, Media Configuration features must be implemented successfully in order to execute the Real Time Streaming test cases.

NVC user must explicitly specify the optional transport protocols supported by DUT.

NVC and DUT time should be synchronized for media streaming.

Real time streaming testing will test only one media stream at a time.

Poor streaming test is out side the scope of the ONVIF Test Specification

Please refer Annex A.3 for the correct interpretation of StreamSetup syntax

Please refer Section 5 for Real Time Streaming Test Cases.

## 4 Media Configuration Test Cases

### 4.1 Media Profile

#### 4.1.1 MEDIA PROFILE CONFIGURATION

**Test Label:** Media Configuration Media Profiles.

**Test Case ID:** MEDIA-1-1-1

**ONVIF Core Specification Coverage:** Get media profiles

**Command Under Test:** GetProfiles

**WSDL Reference:** media.wsdl

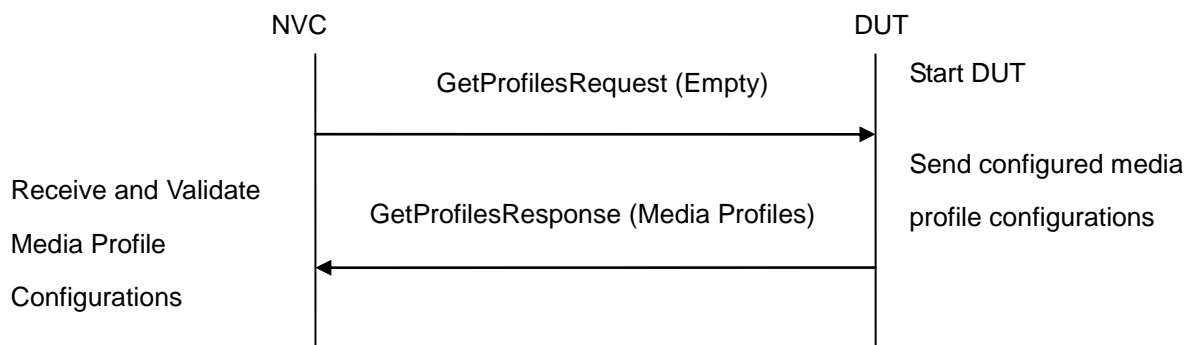
**Requirement Level:** MUST

**Test Purpose:** To retrieve existing media profiles of DUT and the corresponding media entities (video source and video encoder).

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start the DUT.
3. NVC will invoke GetProfilesRequest message to retrieve existing media profiles of the DUT.
4. Verify that the DUT returns at least one media profile with video configuration (video source and video encoder) in the GetProfilesResponse message.
5. Verify that all media profile elements have 'fixed' attribute.



**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send GetProfilesResponse message.

DUT has no default media profile configuration.

DUT did not provide at least one media profile with video source and video encoder configurations.

One or more media profiles don't have 'fixed' attribute.

#### **4.1.2 DYNAMIC MEDIA PROFILE CONFIGURATION**

**Test Label:** Media Configuration Dynamic Media Profile Configuration.

**Test Case ID:** MEDIA-1-1-2

**ONVIF Core Specification Coverage:** Create media profile, Get media profiles , Get media profile, Add video source configuration to a profile, Add video encoder configuration to a profile, Remove video source configuration from a profile, Remove video encoder configuration from a profile, Delete media profile.

**Command Under Test:** None

**WSDL Reference:** media.wsdl

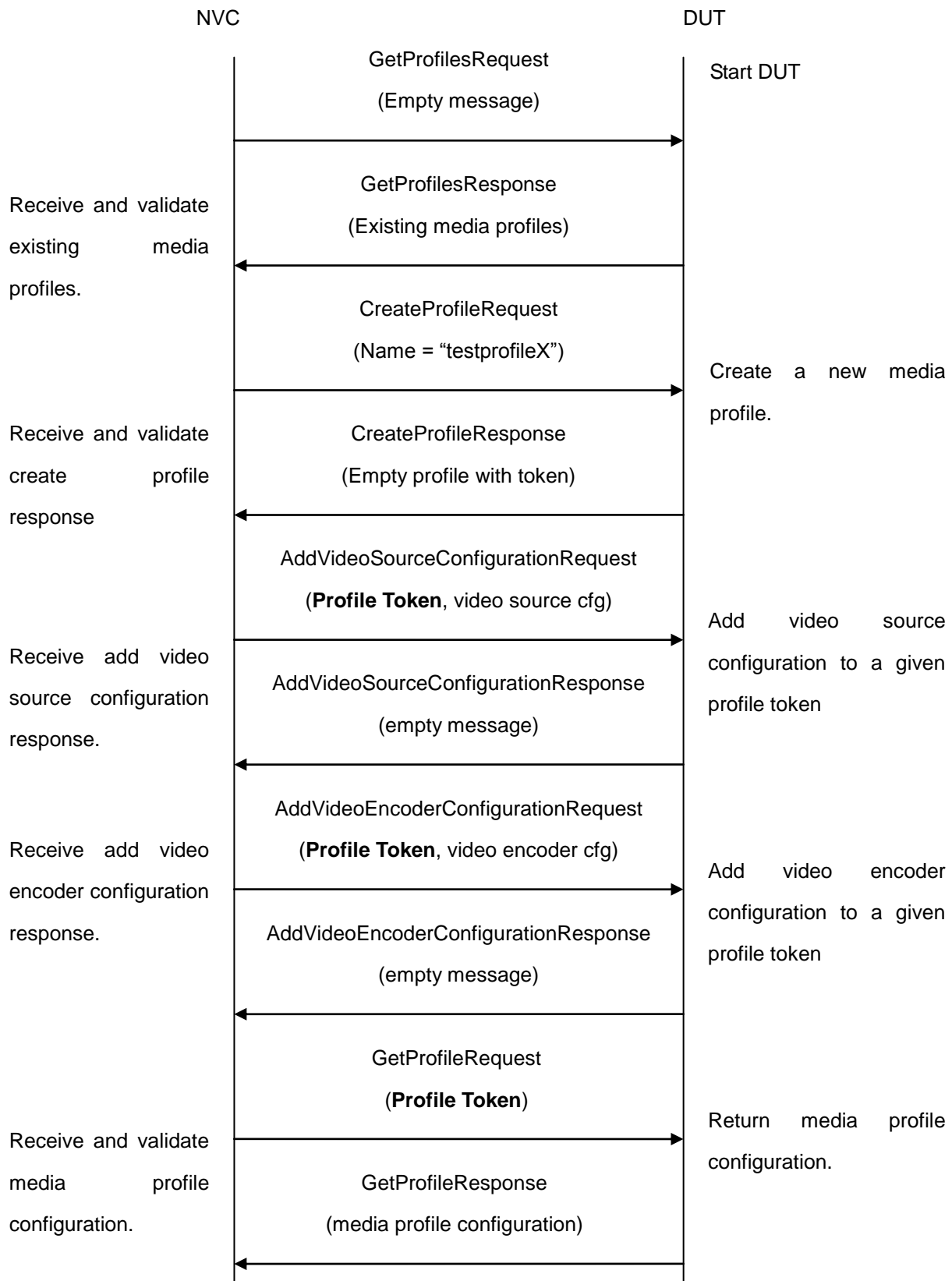
**Requirement Level:** MUST

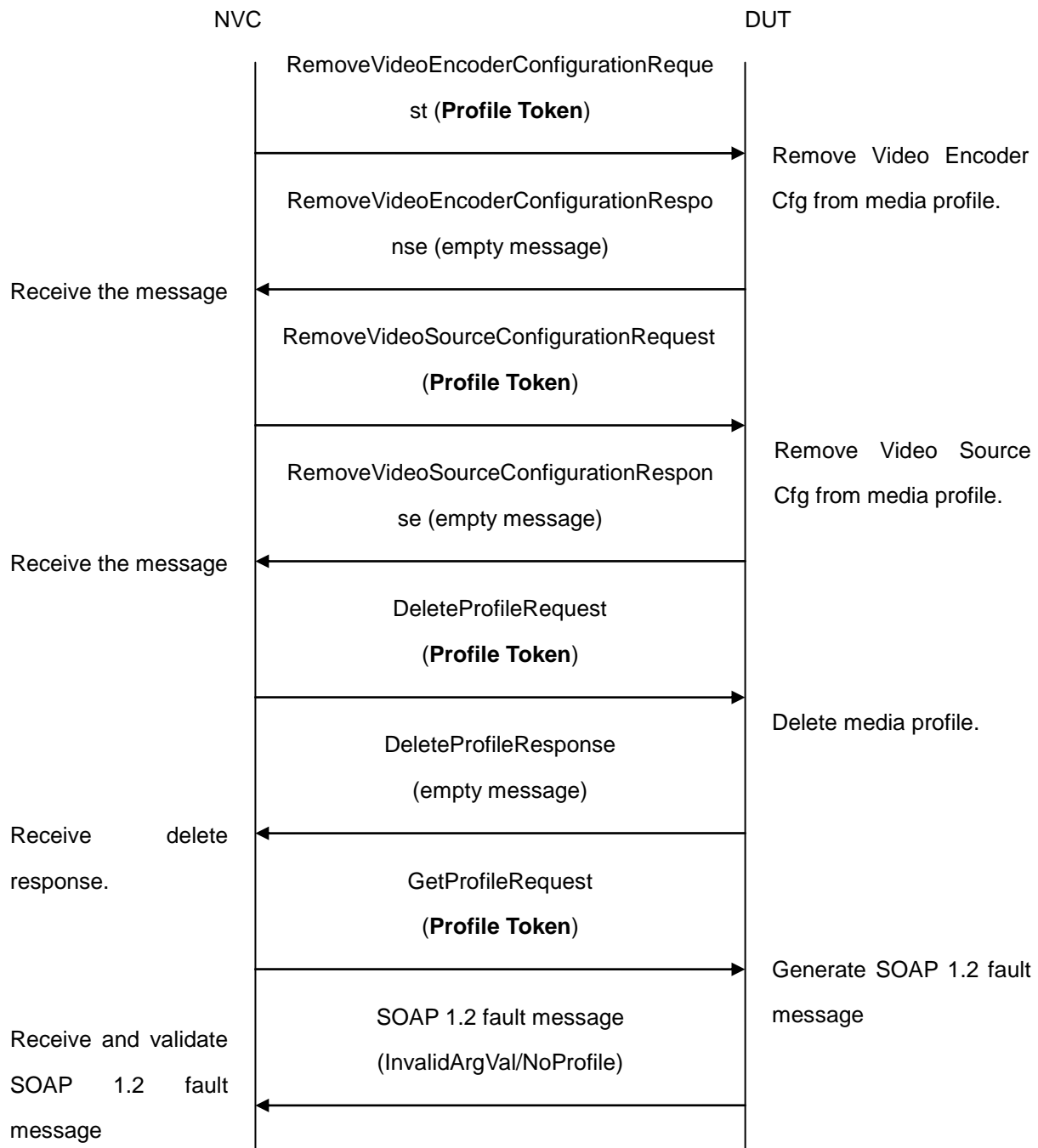
**Test Purpose:** To verify the behaviour of the DUT for dynamic media profile configuration.

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC will invoke GetProfilesRequest message to retrieve existing media profiles configurations of the DUT.

4. Verify that the DUT returns at-least one media profile with video configuration (video source and video encoder) in GetProfilesResponse message.
5. NVC will invoke CreateProfileRequest message to create a new empty media profile.
6. DUT returns an empty profile with no profile entities in the CreateProfileResponse message.
7. NVC will invoke AddVideoSourceConfigurationRequest message (**Profile Token**, Reference to **Video Source Configuration of existing media profile**) to add video source configuration to new profile.
8. DUT sends AddVideoSourceConfigurationResponse message indicating successful addition of video source configuration.
9. NVC will invoke AddVideoEncoderConfigurationRequest message (**Profile Token**, Reference to **Video Encoder Configuration of existing media profile**) to add video encoder configuration to new profile.
10. DUT sends AddVideoEncoderConfigurationResponse message indicating successful addition of video encoder configuration.
11. NVC will invoke GetProfileRequest (**Profile Token**) message to verify video source and encoder configurations in a new profile.
12. DUT will return media profile configuration for requested media profile in the GetProfileResponse message.
13. NVC will invoke RemoveVideoEncoderConfigurationRequest message (**Profile Token**) to remove video encoder configuration from a media profile.
14. DUT sends RemoveVideoEncoderConfigurationResponse message indicating successfully removal of video encoder configuration.
15. NVC will invoke RemoveVideoSourceConfigurationRequest message (**Profile Token**) to remove video source configuration from a media profile.
16. DUT sends RemoveVideoSourceConfigurationResponse message indicating successfully removal of video source configuration.
17. NVC will invoke DeleteProfileRequest (**Profile Token**) message to delete the newly created media profile.
18. DUT will delete the media profile and sends DeleteProfileResponse message.
19. NVC will invoke GetProfileRequest (**deleted Profile Token**) message to check the existence of deleted media profile.
20. DUT will generate SOAP 1.2 fault message (**InvalidArgVal/NoProfile**).

#### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send CreateProfileResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send GetProfileResponse message.

The DUT did not set 'fixed' attribute of created media profile to 'false'

The DUT did not send RemoveVideoEncoderConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

The DUT did not send SOAP 1.2 fault message (InvalidArgVal/NoProfile).

**Note:** See Annex in [ONVIF Base Test] for Invalid SOAP 1.2 fault message definition.

#### 4.1.3 PROFILES CONSISTENCY

**Test Label:** Media Service DUT GetProfiles Command and GetProfile Command Consistency Validation.

**Test Case ID:** MEDIA-1-1-3

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile

**Command Under Test:** GetProfiles, GetProfile

**WSDL Reference:** media.wsdl

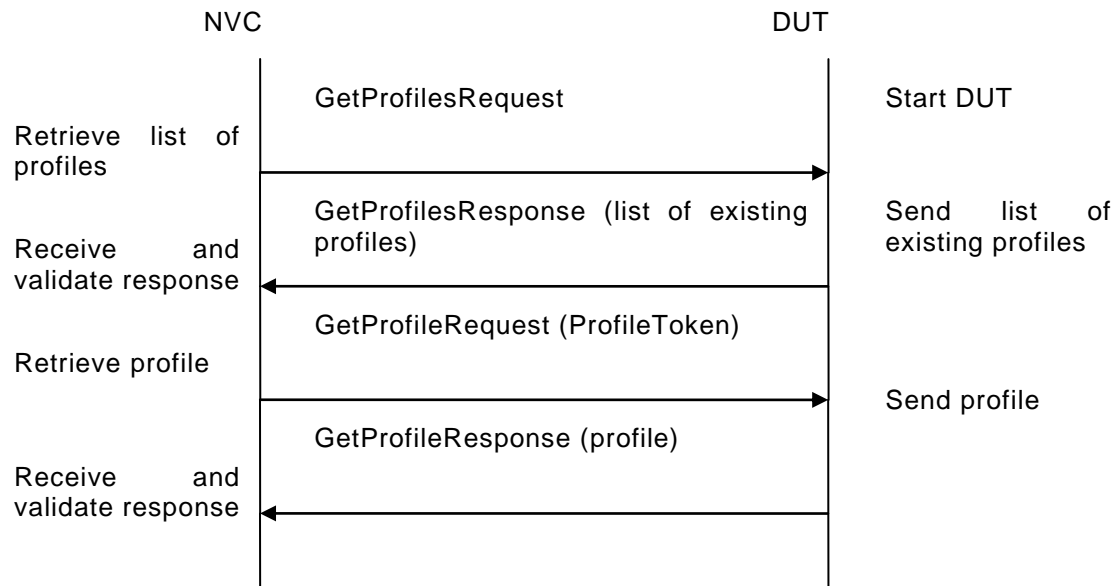
**Requirement Level:** MUST

**Test Propose:** To check that GetProfile command and GetProfiles command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetProfilesRequest` message to retrieve list of profiles and their Configurations
4. The DUT sends `GetProfilesResponse` message.
5. Check that each Profile from `GetProfilesResponse` message has unique token.
6. NVC invokes `GetProfileRequest (ProfileToken)` message to retrieve profile from device.
7. The DUT sends `GetProfileResponse` message.
8. Verify that all parameters and their values for the same profile from `GetProfilesResponse` message and `GetProfileResponse` message are same.
9. Repeat steps 6-8 for other profiles from the `GetProfilesResponse` message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send `GetProfilesResponse` message.

The DUT did not send valid `GetProfilesResponse` message.

The DUT return two or more Profiles in `GetProfilesResponse` message with the same `ProfileToken`.

The DUT did not send `GetProfileResponse` message.

The DUT did not send valid GetProfileResponse message.

The DUT did not send equal parameters in the GetProfileResponse message and in the GetProfilesResponse message for the same profile.

## 4.2 Video Configuration

### 4.2.1 VIDEO SOURCE CONFIGURATION

**Test Label:** Media Configuration Video Source Configuration

**Test Case ID:** MEDIA-2-1-1

**ONVIF Core Specification Coverage:** Get media profiles, GetVideoSources, Get video source configurations, Get video source configuration, Get compatible video source configurations, Get video source configuration options, Modify a video source configuration

**Command Under Test:** None

**WSDL Reference:** media.wsdl

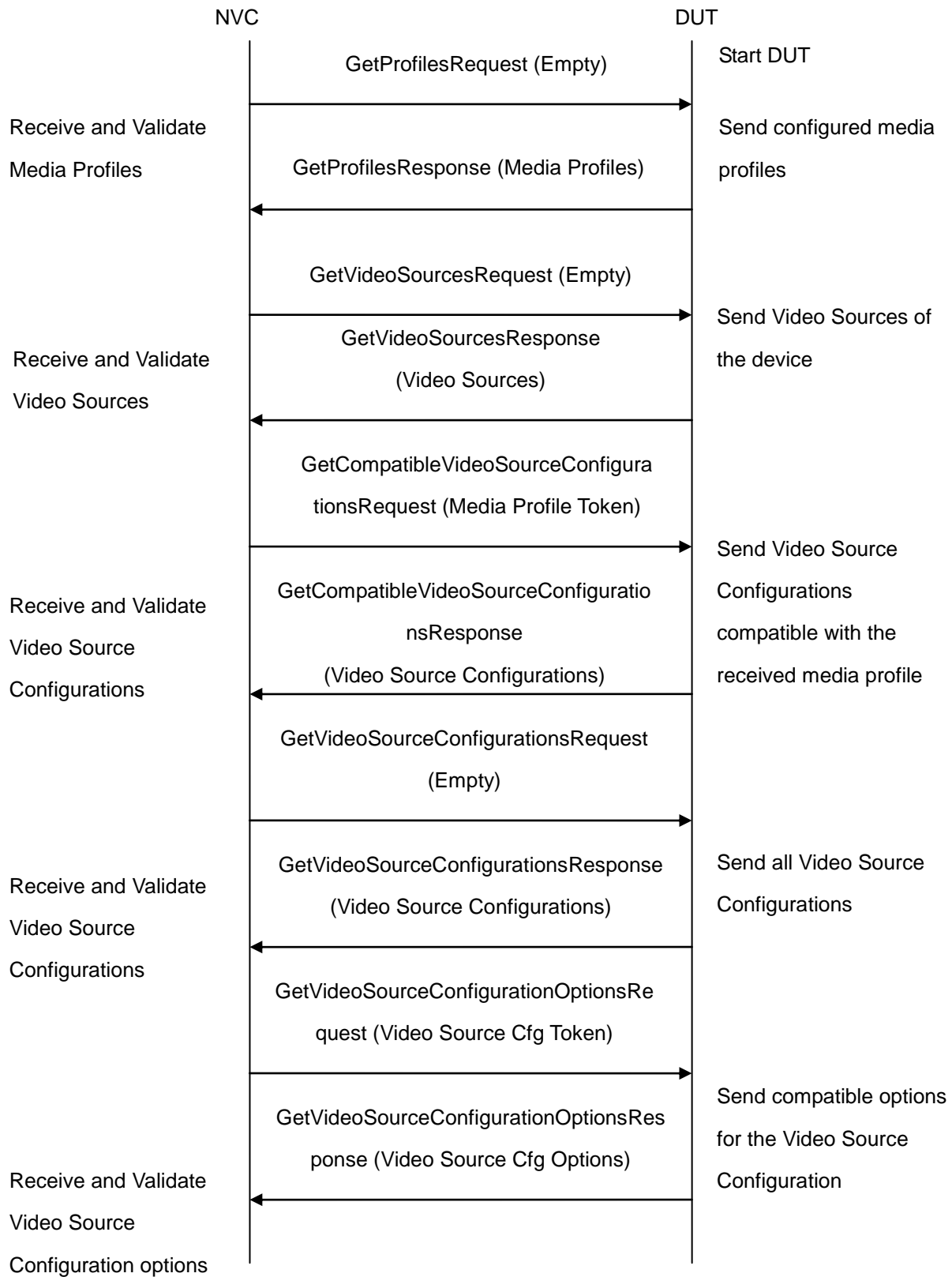
**Requirement Level:** MUST

**Test Purpose:** To verify DUT Video Source Configuration Operations

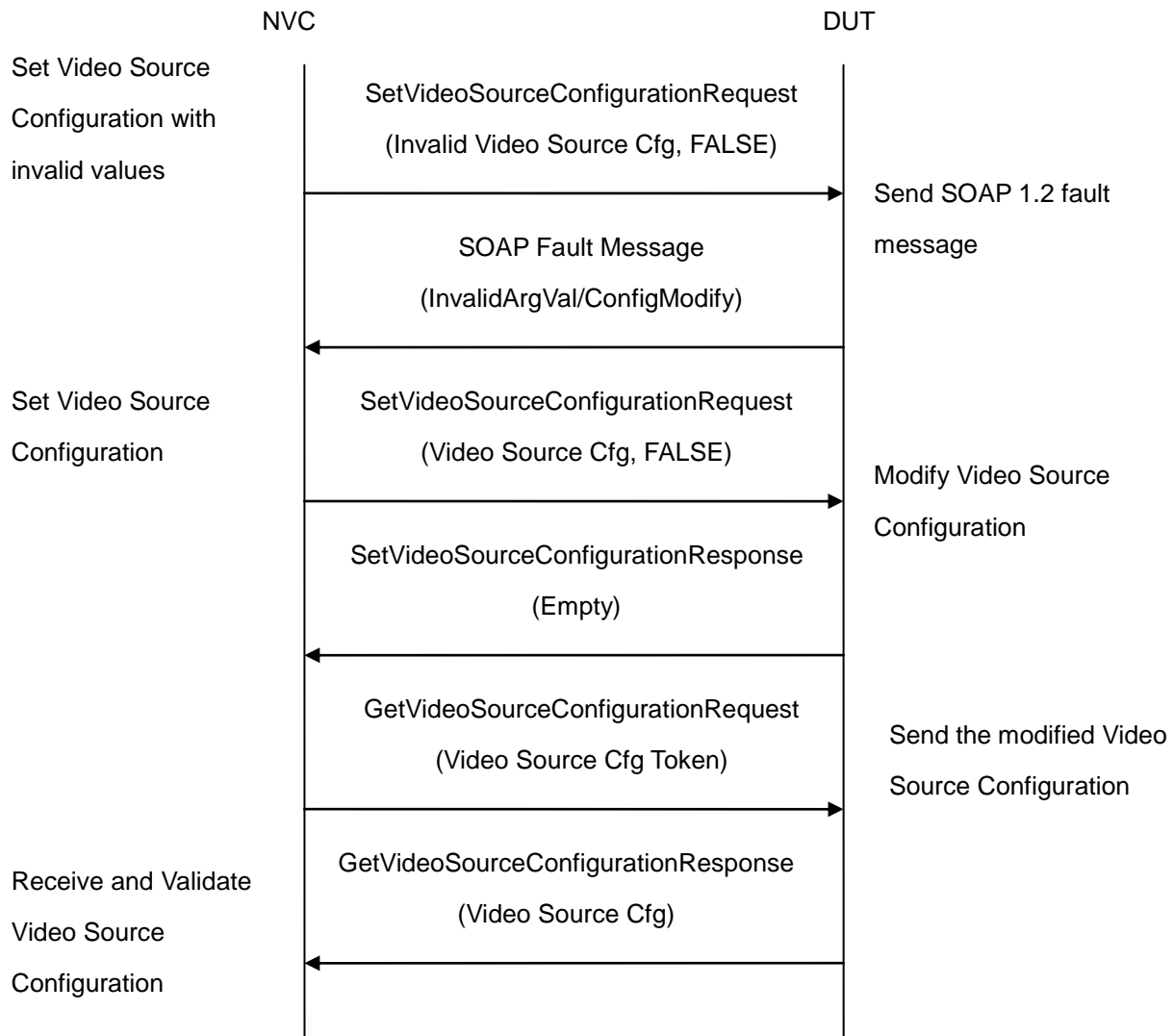
**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes GetProfiles Request.
4. DUT sends the list of existing media profiles in GetProfiles Response message.
5. NVC invokes GetVideoSources request to retrieve the existing video sources of DUT.
6. NVC verifies the list of video sources sent by DUT.
7. NVC invokes GetCompatibleVideoSourceConfigurations request with one of the received media profile tokens as input argument.

8. DUT sends the list of video source configurations compatible with the received media profile token.
9. NVC verifies the list of video source configurations sent by DUT.
10. NVC invokes `GetVideoSourceConfigurations` request to retrieve the list of video source configurations supported by the DUT.
11. NVC verifies the list of video source configurations sent by the DUT.
12. NVC invokes `GetVideoSourceConfigurationOptions` request with one of the received video source configuration tokens as input argument.
13. DUT sends the range of configurable values supported by it for the received video source configuration token.
14. NVC invokes `SetVideoSourceConfiguration` request with video source configuration values outside the range specified in the `GetVideoSourceConfigurationOptionsResponse` and '**ForcePersistence**' flag as '**FALSE**'.
15. DUT send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**).
16. NVC verifies the SOAP fault message sent by DUT.
17. NVC invokes `SetVideoSourceConfiguration` request with video source configuration values within the range specified in `GetVideoSourceConfigurationOptionsResponse` and '**ForcePersistence**' flag as '**FALSE**'.
18. DUT modifies the video source configuration and sends the `SetVideoSourceConfiguration` response indicating success.
19. NVC invokes `GetVideoSourceConfiguration` request to verify the modified video source configuration.
20. DUT sends the modified video source configuration in `GetVideoSourceConfiguration` response.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send `GetProfilesResponse` message.

DUT did not send valid `GetVideoSourcesResponse` message.

DUT did not send valid `GetCompatibleVideoSourceConfigurationsResponse` message.

DUT did not send valid `GetVideoSourceConfigurationsResponse` message.

DUT did not send `GetVideoSourceConfigurationOptionsResponse` message.

DUT did not send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**) for invalid `SetVideoSourceConfiguration` request.

DUT did not send `SetVideoSourceConfigurationResponse` message.

DUT did not send GetVideoSourceConfigurationResponse message.

DUT did not modify video source configuration correctly.

#### **4.2.2 VIDEO ENCODER CONFIGURATION**

**Test Label:** Media Configuration Video Encoder Configuration

**Test Case ID:** MEDIA-2-1-2

**ONVIF Core Specification Coverage:** Get media profiles, Get video encoder configurations, Get compatible video encoder configurations

**Command Under Test:** None

**WSDL Reference:** media.wsdl

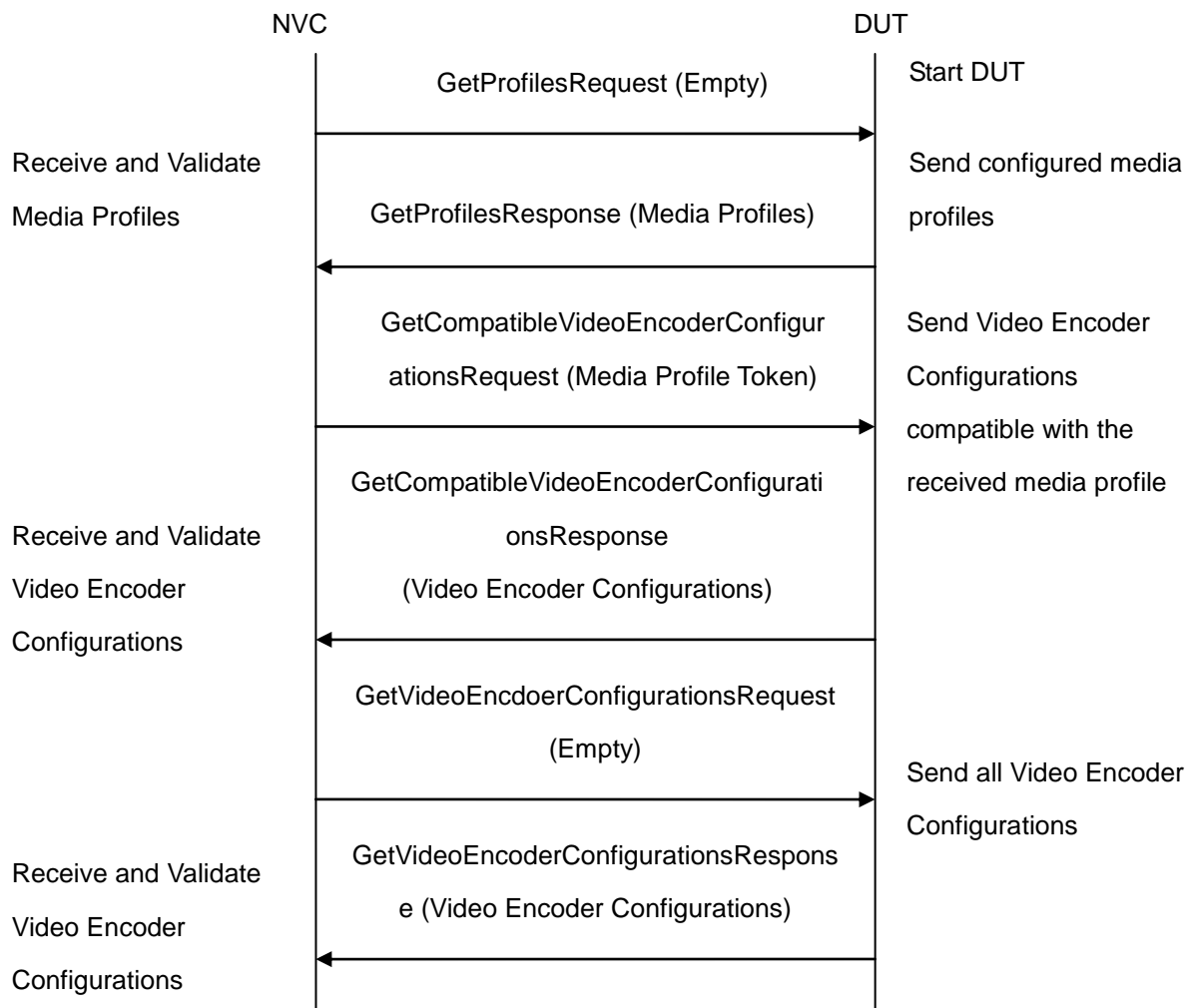
**Requirement Level:** MUST

**Test Purpose:** To verify DUT Video Encoder Configuration Operations

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes GetProfiles Request.
4. DUT sends the list of existing media profiles in GetProfiles Response message.
5. NVC invokes GetCompatibleVideoEncoderConfigurations request with one of the received media profile tokens as input argument.
6. DUT sends the list of video encoder configurations, compatible with the received media profile token.
7. NVC verifies the list of video encoder configurations sent by DUT.
8. NVC will invoke GetVideoEncoderConfigurations request to retrieve the list of video encoder configurations supported by the DUT.
9. DUT sends the list of all video encoder configurations supported by it.

10. NVC verifies the list of video encoder configurations sent by the DUT.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send GetProfilesResponse message.

DUT did not send valid GetCompatibleVideoEncoderConfigurationsResponse message.

DUT did not send valid GetVideoEncoderConfigurationsResponse message.

#### 4.2.3 JPEG VIDEO ENCODER CONFIGURATION

**Test Label:** Media Configuration JPEG Video Encoder Configuration

**Test Case ID:** MEDIA-2-1-3

**ONVIF Core Specification Coverage:** Get video encoder configurations, Get video encoder configuration, Get video encoder configuration options, Modify a video encoder configuration.

**Command Under Test:** None

**WSDL Reference:** media.wsdl

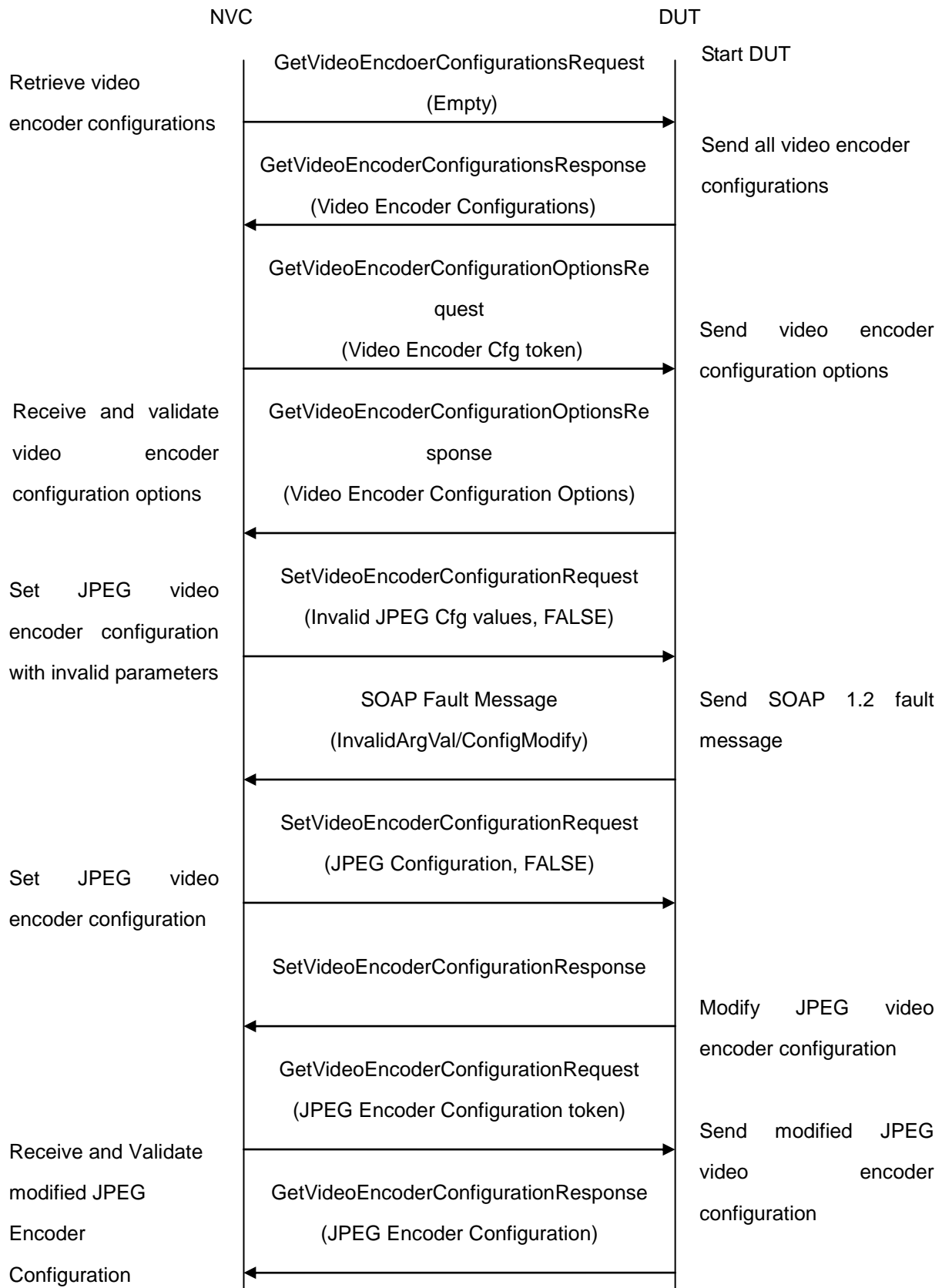
**Requirement Level:** MUST

**Test Purpose:** To verify DUT JPEG Video Encoder Configurations Setting

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start the DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` to retrieve the list of video encoder configurations supported by DUT.
4. DUT sends video encoder configurations in the `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoEncoderConfigurationOptions Request` (Video Encoder Configuration token) to retrieve video encoder configuration options for the specified video encoder configuration.
6. DUT sends the range of configurable values for the received video encoder configuration in the `GetVideoEncoderConfigurationOptionsResponse` message.
7. Test steps -5 & 6 have to be repeated for all video encoder configurations until NVC finds a video encoder configuration with JPEG encoding support.
8. NVC invokes `SetVideoEncoderConfiguration` request with JPEG configuration values outside the range defined in the `GetVideoEncoderConfigurationOptionsResponse` and 'ForcePersistence' flag as 'FALSE'.
9. DUT send the SOAP 1.2 fault message (`InvalidArgVal/ConfigModify`)
10. NVC verifies the SOAP fault message sent by DUT.
11. NVC invokes `SetVideoEncoderConfiguration` request (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1 and force persistence = false**). These values will be taken from `GetVideoEncoderConfigurationOptionsResponse` message.
12. DUT modifies JPEG video encoder configuration and responds with `SetVideoEncoderConfigurationResponse` message indicating success.
13. NVC verifies the JPEG Video Encoder Configuration settings on DUT by invoking `GetVideoEncoderConfiguration` request.
14. DUT sends modified JPEG Video Encoder Configuration in the `GetVideoEncoderConfigurationResponse` message (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1**).

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send `GetVideoEncoderConfigurationsResponse` message.

DUT did not send `GetVideoEncoderConfigurationOptionsResponse` message.

DUT doesn't support JPEG encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetVideoEncoderConfiguration request.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not modify JPEG Video Encoder Configuration.

#### **4.2.4 MPEG4 VIDEO ENCODER CONFIGURATION**

**Test Label:** Media Configuration MPEG4 Video Encoder Configuration

**Test Case ID:** MEDIA-2-1-4

**ONVIF Core Specification Coverage:** Get video encoder configurations, Get video encoder configuration, Get video encoder configuration options, Modify a video encoder configuration.

Command Under Test: None

WSDL Reference: media.wsdl

**Requirement Level:** MUST IF IMPLEMENTED (MPEG4-SP)

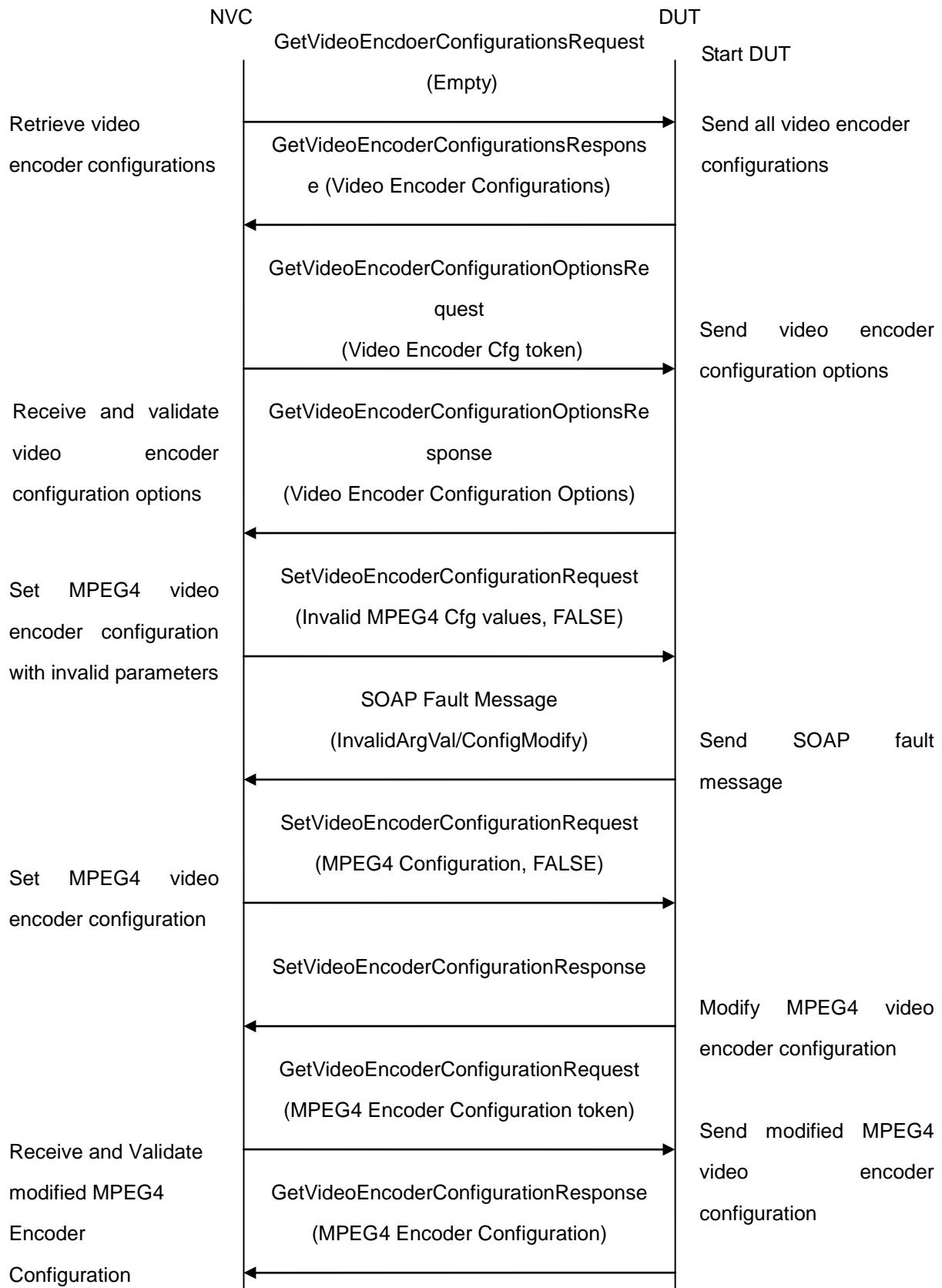
**Test Purpose:** To verify DUT MPEG4 Video Encoder Configurations Setting

**Pre-Requisite:** MPEG4 is implemented by DUT

Test Configuration: NVC and DUT

Test Sequence:





#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` to retrieve the list of video encoder configurations supported by DUT.
4. DUT sends video encoder configurations in the `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoEncoderConfigurationOptions Request` (Video Encoder Configuration token) to retrieve video encoder configuration options for the specified video encoder configuration.
6. DUT sends the range of configurable values for the received video encoder configuration in the `GetVideoEncoderConfigurationOptionsResponse` message.
7. Test steps -5 & 6 have to be repeated for all video encoder configurations until NVC finds a video encoder configuration with MPEG4 encoding support.
8. NVC invokes `SetVideoEncoderConfiguration` request with MPEG4 configuration values outside the range defined in the `GetVideoEncoderConfigurationOptionsResponse` and 'ForcePersistence' flag as 'FALSE'.
9. DUT send the SOAP 1.2 fault message (`InvalidArgVal/ConfigModify`)
10. NVC verifies the SOAP fault message sent by DUT.
11. NVC invokes `SetVideoEncoderConfiguration` request (Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, Mpeg4Profile = SP and force persistence = false). These values will be taken from `GetVideoEncoderConfigurationOptionsResponse` message.
12. DUT modifies MPEG4 video encoder configuration and responds with `SetVideoEncoderConfigurationResponse` message indicating success.
13. NVC verifies the MPEG4 Video Encoder Configuration settings on DUT by `GetVideoEncoderConfigurationRequest` message.
14. DUT sends modified MPEG4 Video Encoder Configuration in the `GetVideoEncoderConfigurationResponse` message (**Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, Mpeg4Profile = SP**).

#### Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send `GetVideoEncoderConfigurationsResponse` message.

DUT did not send `GetVideoEncoderConfigurationOptionsResponse` message.

DUT doesn't support MPEG4 encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetVideoEncoderConfiguration request.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not modify MPEG4 Video Encoder Configuration.

#### 4.2.5 H.264 VIDEO ENCODER CONFIGURATION

**Test Label:** Media Configuration H.264 Video Encoder Configuration

**Test Case ID:** MEDIA-2-1-5

**ONVIF Core Specification Coverage:** Get video encoder configurations, Get video encoder configuration, Get video encoder configuration options, Modify a video encoder configuration.

Command Under Test: None

WSDL Reference: media.wsdl

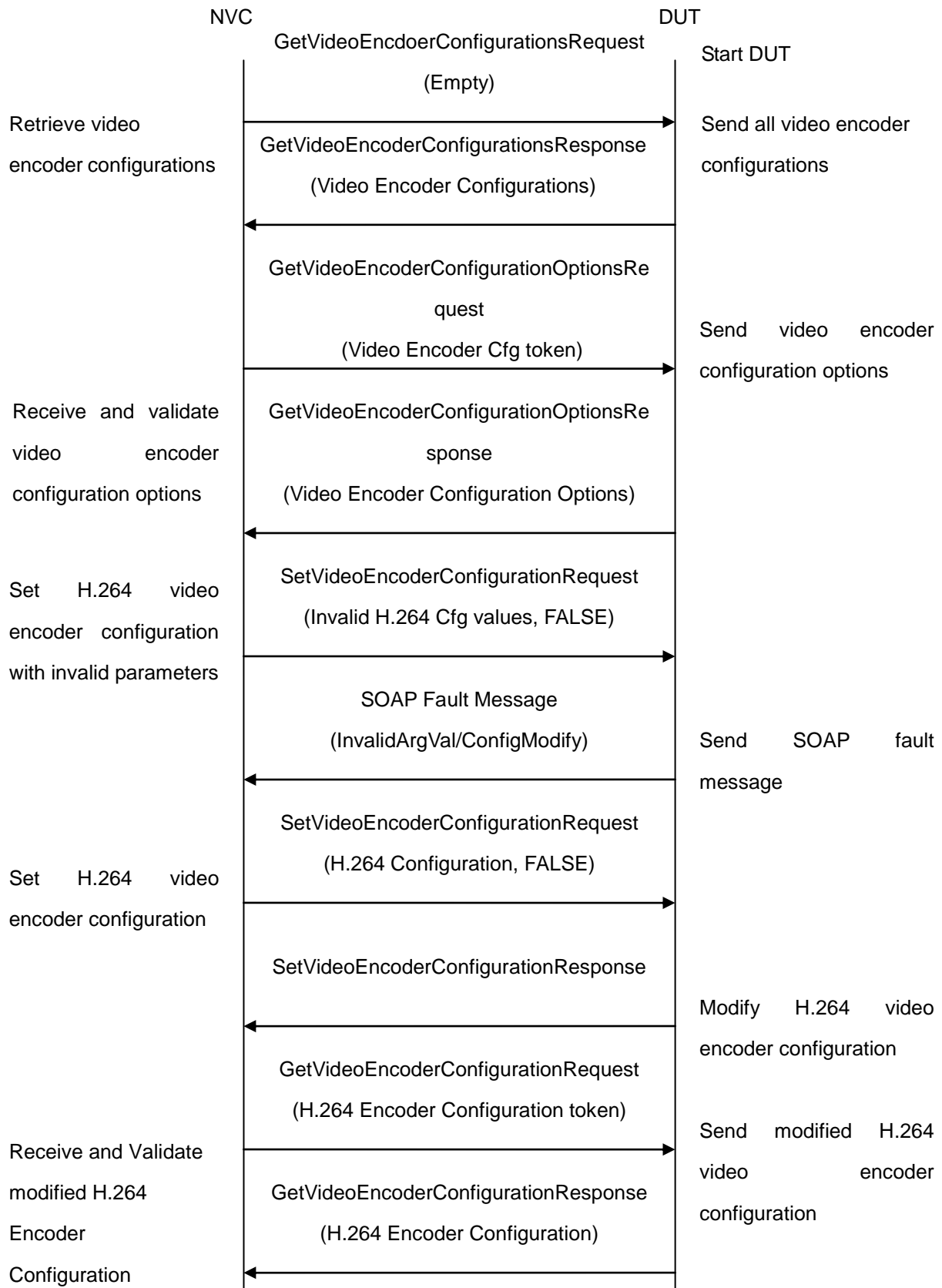
**Requirement Level:** MUST IF IMPLEMENTED (H.264-Baseline)

**Test Purpose:** To verify DUT H.264 Video Encoder Configurations Setting

**Pre-Requisite:** H.264 is implemented by DUT

Test Configuration: NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` to retrieve the list of video encoder configurations supported by DUT.
4. DUT sends video encoder configurations in the `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoEncoderConfigurationOptions` Request (Video Encoder Configuration token) to retrieve video encoder configuration options for the specified video encoder configuration.
6. DUT sends the range of configurable values for the received video encoder configuration in the `GetVideoEncoderConfigurationOptionsResponse` message.
7. Test steps -5 & 6 have to be repeated for all video encoder configurations until NVC finds a video encoder configuration with H.264 encoding support
8. NVC invokes `SetVideoEncoderConfiguration` request with H.264 configuration values outside the range defined in the `GetVideoEncoderConfigurationOptionsResponse` and 'ForcePersistence' flag as 'FALSE'.
9. DUT send the SOAP 1.2 fault message (`InvalidArgVal/ConfigModify`)
10. NVC verifies the SOAP fault message sent by DUT.
11. NVC will invoke `SetVideoEncoderConfiguration` request (`Encoding = "H.264"`, `Resolution = ["Width", "Height"]`, `Quality = q1`, `GovLength = g1`, `H264Profile = "Baseline"` and force persistence = false). These values will be taken from `GetVideoEncoderConfigurationOptionsResponse` message.
12. DUT modifies H.264 video encoder configuration and responds with `SetVideoEncoderConfigurationResponse` message indicating success.
13. NVC will verify the H.264 Video Encoder Configuration settings on DUT by `GetVideoEncoderConfigurationRequest` message.
14. DUT sends modified H.264 Video Encoder Configurations in the `GetVideoEncoderConfigurationResponse` message (**`Encoding = "H.264"`, `Resolution = ["Width", "Height"]`, `Quality = q1`, `GovLength = g1`, `H264Profile = "Baseline"`**).

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not send `GetVideoEncoderConfigurationsResponse`.

DUT did not send `GetVideoEncoderConfigurationOptionsResponse` message.

DUT doesn't support H.264 encoding.

DUT did not send the SOAP 1.2 fault message (`InvalidArgVal/ConfigModify`) for invalid `SetVideoEncoderConfiguration` request.

DUT did not send GetVideoEncoderConfigurationResponse message.

DUT did not send SetVideoEncoderConfigurationResponse message.

The DUT did not modify H.264 Video Encoder Configuration.

#### **4.2.6 GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES**

**Test Label:** Media Configuration DUT Video Encoder Instances

**Test Case ID:** MEDIA-2-1-6

**ONVIF Core Specification Coverage:** Get video source configurations, Get guaranteed number of video encoder instances

**Command Under Test:** GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

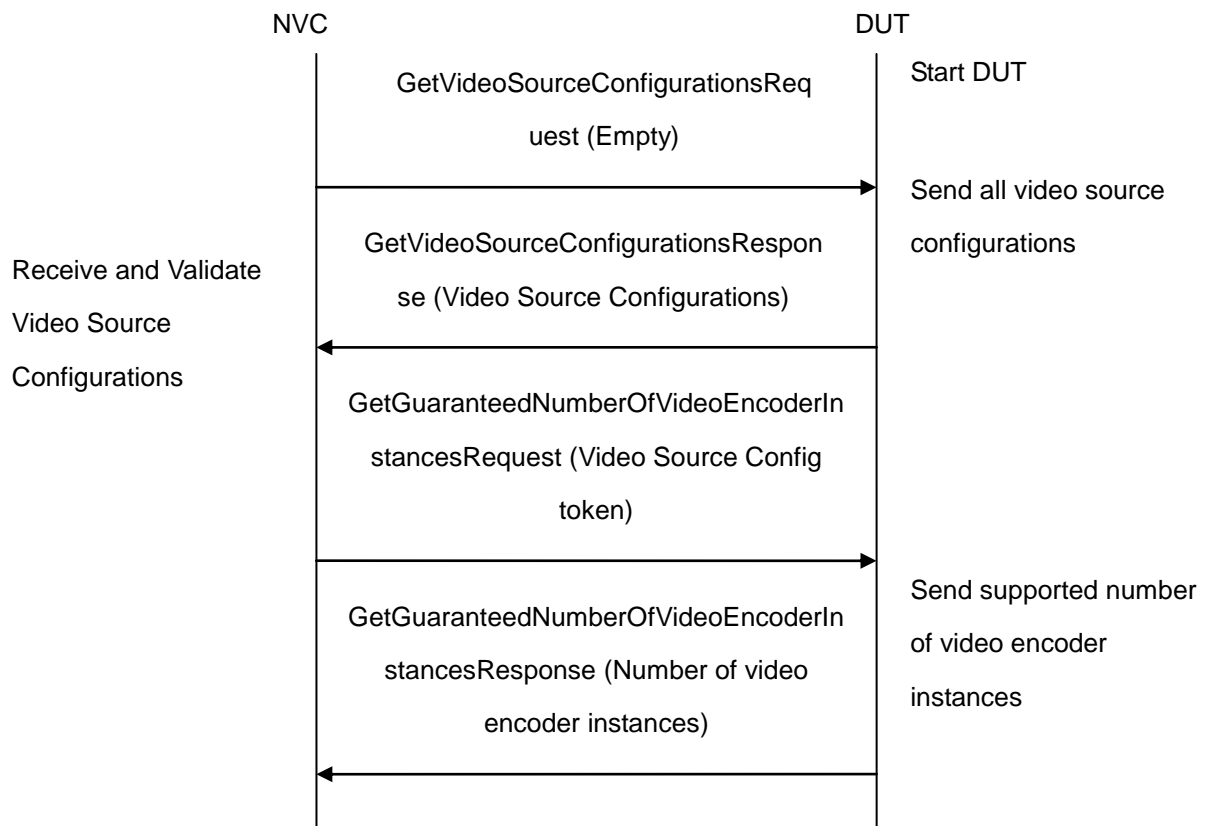
**Requirement Level:** MUST

**Test Purpose:** To retrieve minimum number of video encoder instances supported by DUT per video source configuration.

Pre-Requisite: None

Test Configuration: NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes `GetVideoSourceConfigurations` request.
4. DUT sends the list of video source configurations supported by it.
5. NVC invokes `GetGuaranteedNumberOfVideoEncoderInstancesRequest` message for a selected video source configuration.
6. DUT sends the minimum guaranteed number of video encoder instances.

#### Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send `GetVideoSourceConfigurationsResponse` message.

DUT did not send `GetGuaranteedNumberOfVideoEncoderInstancesResponse` message.

DUT did not send 'TotalNumber' value.

### 4.3 Video Source Configuration

#### 4.3.1 VIDEO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY

**Test Label:** Media Service DUT GetVideoSourceConfigurations Command and GetProfiles Command Consistency Validation.

**Test Case ID:** MEDIA-2-2-1

**ONVIF Core Specification Coverage:** Get media profiles, Get video source configurations

**Command Under Test:** GetProfiles, GetVideoSourceConfigurations

WSDL Reference: media.wsdl

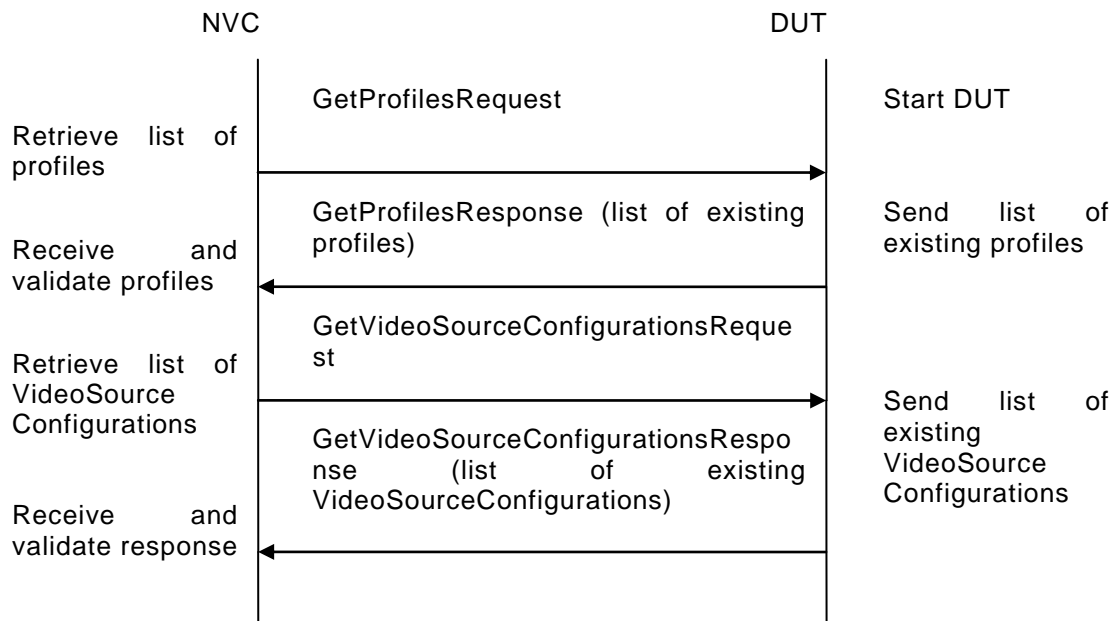
**Requirement Level:** MUST

**Test Propose:** To check that GetVideoSourceConfigurations command and GetProfiles command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

Test Configuration: NVC and DUT

Test Sequence:



Test Procedure:

1. Start an NVC.
2. Start a DUT.



3. NVC invokes GetProfilesRequest message to retrieve list of profiles and their configurations.
4. The DUT sends GetProfilesResponse message.
5. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.
6. The DUT sends GetVideoSourceConfigurationsResponse message.
7. Check that each VideoSourceConfiguration from GetVideoSourceConfigurationsResponse message has unique token.
8. Check that each VideoSourceConfigurations from the GetProfilesResponse message are included in the GetVideoSourceConfigurationsResponse message.
9. Check that VideoSourceConfiguration parameters are same in the GetProfilesResponse message and in the GetVideoSourceConfigurationsResponse message for each VideoSourceConfiguration.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT return two or more VideoSourceConfigurations in GetVideoSourceConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with VideoSourceConfigurations that were not included in the GetVideoSourceConfigurationsResponse message.

The DUT returned different parameters list and parameters values for the same VideoSourceConfiguration in the GetVideoSourceConfigurationsResponse message and in the GetProfilesResponse message.

#### 4.3.2 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION CONSISTENCY

**Test Label:** Media Service DUT GetVideoSourceConfigurations Command and GetVideoSourceConfiguration Command Consistency Validation.

**Test Case ID:** MEDIA-2-2-2

**ONVIF Core Specification Coverage:** Get video source configurations, Get video source configuration

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSourceConfiguration

WSDL Reference: media.wsdl

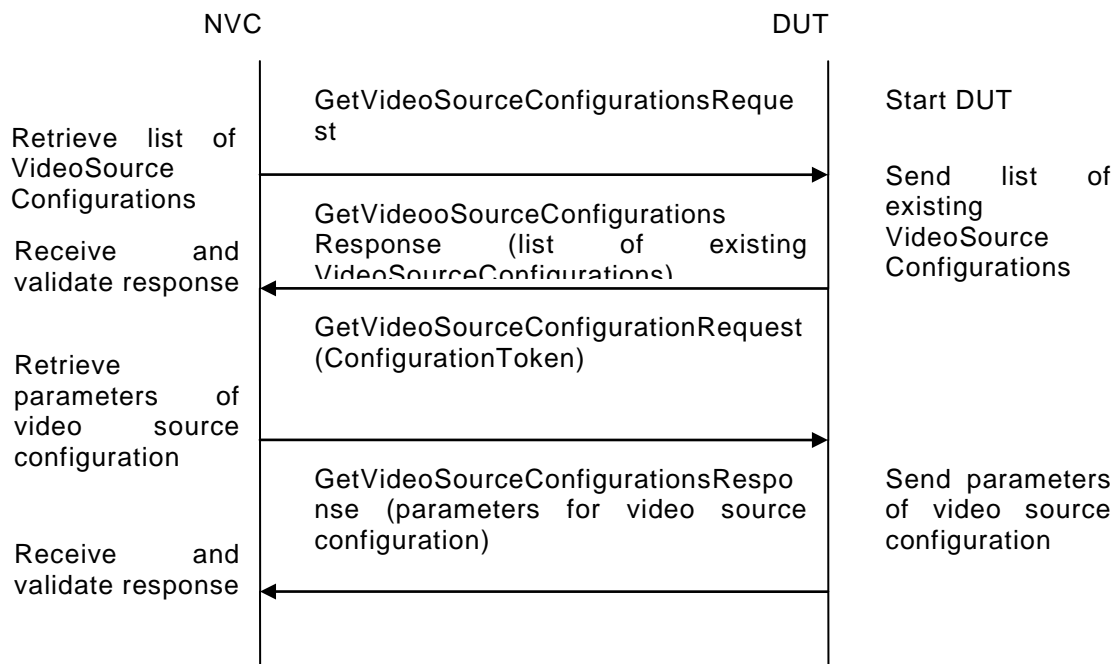
### Requirement Level: MUST

**Test Propose:** To check that GetVideoSourceConfigurations and GetVideoSourceConfiguration are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

Test Configuration: NVC and DUT

Test Sequence:



Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.
4. The DUT sends GetVideoSourceConfigurationsResponse message.
5. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken**) message to retrieve parameters of Video Source Configuration from device.
6. The DUT sends GetVideoSourceConfigurationResponse message.
7. Verify that all parameters and their values for video source ConfigurationToken from GetVideoSourceConfigurationsResponse message and GetVideoSourceConfigurationResponse message are the same.
8. Repeat steps 5-7 for other VideoSourceConfigurations from the GetVideoSourceConfigurationsResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

The DUT did not send equal parameters for VideoSourceConfiguration in the GetVideoSourceConfigurationResponse message and in the GetVideoSourceConfigurationsResponse message.

### 4.3.3 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION OPTIONS

#### CONSISTENCY

**Test Label:** Media Service DUT GetVideoSourceConfigurations Command and GetVideoSourceConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-2-2-3

**ONVIF Core Specification Coverage:** Get video source configuration options, Get video source configurations

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSourceConfigurationOptions

WSDL Reference: media.wsdl

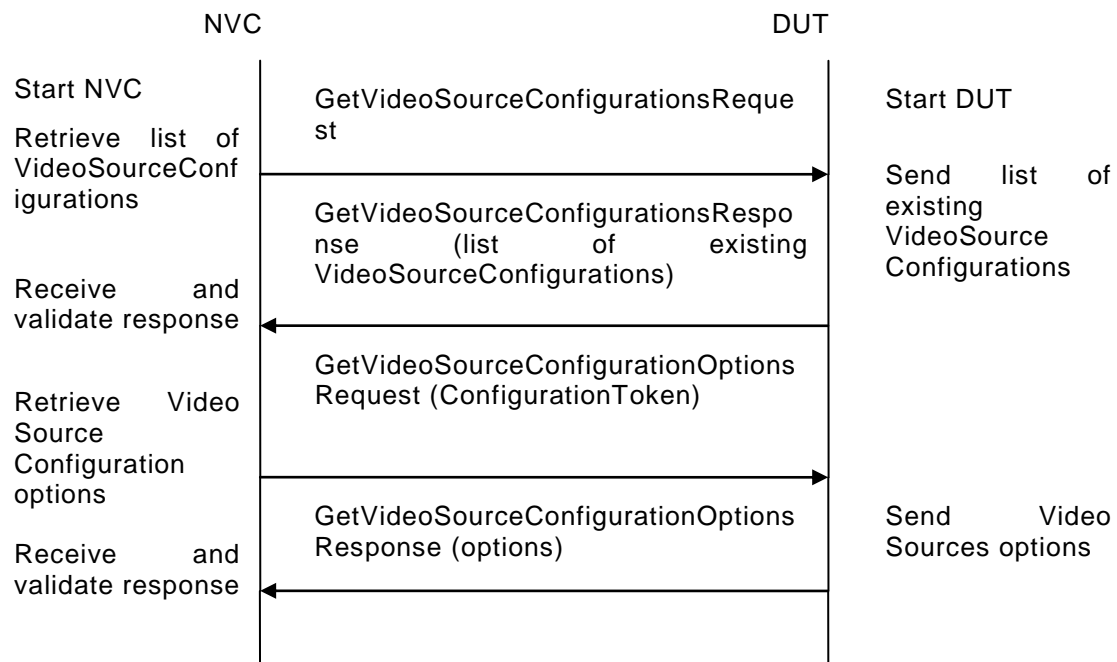
**Requirement Level:** MUST

**Test Propose:** To check that GetVideoSourceConfigurations Command and GetVideoSourceConfigurationOptions Command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

Test Configuration: NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.
4. The DUT sends GetVideoSourceConfigurationsResponse message.
5. NVC invokes GetVideoSourceConfigurationOptionsRequest (**ConfigurationToken**) message to retrieve Video Source Configuration Options for video source configuration from device.
6. The DUT sends GetVideoSourceConfigurationOptionsResponse message.
7. Check that parameters for video source configuration are available according to options for video source configuration.
8. Repeat steps 5-7 for each VideoSourceConfiguration from the GetVideoSourceConfigurationsResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourceConfigurationOptionsResponse message.

The DUT did not send valid GetVideoSourceConfigurationOptionsResponse message.

The DUT did send inconsistent GetVideoSourceConfigurationOptionsResponse message:

- Source token of VideoSourceConfiguration does not exist in VideoSourceTokensAvailable list for this VideoSourceConfiguration in GetVideoSourceConfigurationOptionsResponse.
- VideoSourceConfiguration.Bounds.x is not between BoundsRange.XRange.Min and BoundsRange.XRange.Max.
- VideoSourceConfiguration.Bounds.y is not between BoundsRange.YRange.Min and BoundsRange.YRange.Max.
- VideoSourceConfiguration.Bounds.width is not between BoundsRange.WidthRange.Min and BoundsRange.WidthRange.Max.
- VideoSourceConfiguration.Bounds.height is not between BoundsRange.HeightRange.Min and BoundsRange.HeightRange.Max.

#### 4.3.4 PROFILES AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY

**Test Label:** Media Service DUT GetProfiles Command and GerVideoSourceConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-2-2-4

**ONVIF Core Specification Coverage:** Get video source configuration options, Get media profiles

**Command Under Test:** GetProfiles, GetVideoSourceConfigurationOptions

WSDL Reference: media.wsdl

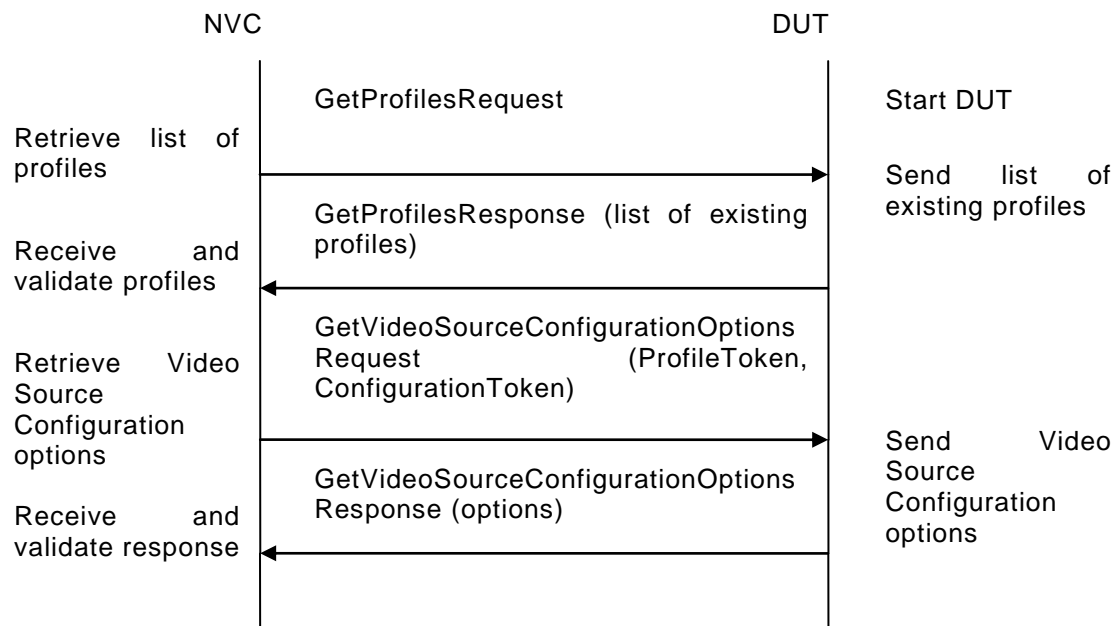
**Requirement Level:** MUST

**Test Propose:** To check that GetProfiles command and GetVideoSourceConfigurationOptions command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

Test Configuration: NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes **GetProfilesRequest** message to retrieve list of profiles from device.
4. The DUT sends **GetProfilesResponse** message.
5. NVC invokes **GetVideoSourceConfigurationOptionsRequest** (**ProfileToken**, **ConfigurationToken**) message to retrieve **VideoSourceConfigurationOptions** for the video source configuration from device.
6. The DUT sends **GetVideoSourceConfigurationOptionsResponse** message.
7. Check that all parameters for the video source configuration from the profile are correct according to options for video source configuration.
8. Repeat steps 5-7 for other profiles with **VideoSourceConfiguration** from the **GetProfilesResponse** message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send **GetProfilesResponse** message.

The DUT did not send valid **GetProfilesResponse** message.

The DUT did not send **GetVideoSourceConfigurationOptionsResponse** message.

The DUT did not send valid **GetVideoSourceConfigurationOptionsResponse** message.

The DUT did send inconsistent GetVideoSourceConfigurationOptionsResponse message:

- Source token of VideoSourceConfiguration does not exist in VideoSourceTokensAvailable list for this VideoSourceConfiguration in GetVideoSourceConfigurationOptionsResponse.
- VideoSourceConfiguration.Bounds.x is not between BoundsRange.XRange.Min and BoundsRange.XRange.Max.
- VideoSourceConfiguration.Bounds.y is not between BoundsRange.YRange.Min and BoundsRange.YRange.Max.
- VideoSourceConfiguration.Bounds.width is not between BoundsRange.WidthRange.Min and BoundsRange.WidthRange.Max.
- VideoSourceConfiguration.Bounds.height is not between BoundsRange.HeightRange.Min and BoundsRange.HeightRange.Max.

#### 4.3.5 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCES CONSISTENCY

**Test Label:** Media Service DUT GetVideoSourceConfigurations Command and GetVideoSources Command Validation.

**Test Case ID:** MEDIA-2-2-5

**ONVIF Core Specification Coverage:** GetVideoSources, Get video source configurations

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSources

WSDL Reference: media.wsdl

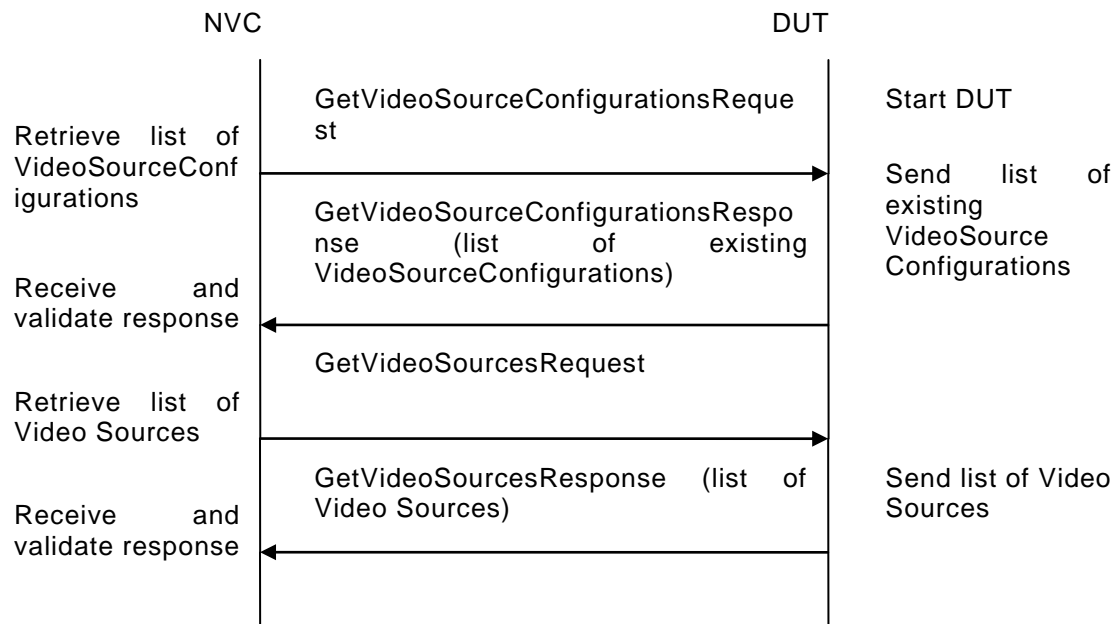
**Requirement Level:** MUST

**Test Propose:** To check that GetVideoSourceConfigurations Command and GetVideoSources Command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

Test Configuration: NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoSourceConfigurationsRequest` message to retrieve list of Video Source Configurations from device.
4. The DUT sends `GetVideoSourceConfigurationsResponse` message.
5. NVC invokes `GetVideoSourcesRequest` message to retrieve list of Video Sources from device.
6. The DUT sends `GetVideoSourcesResponse` message.
7. Check that each `VideoSourceConfiguration` from `GetVideoSourcesResponse` message has unique token.
8. Check that every `VideoSourceConfiguration.SourceToken` from the `GetVideoSourceConfigurationsResponse` message exists in `GetVideoSourcesResponse` message (`VideoSource.token`).

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send `GetVideoSourceConfigurationsResponse` message.

The DUT did not send valid `GetVideoSourceConfigurationsResponse` message.

The DUT did not send `GetVideoSourcesResponse` message.

The DUT did not send valid `GetVideoSourcesResponse` message.



The DUT return two or more VideoSources in GetVideoSourcesResponse message with the same token.

The DUT returned the GetVideoSourceConfigurationsResponse message with VideoSources that were not included in the GetVideoSourceConfigurationsResponse message.

#### **4.3.6 VIDEO SOURCE CONFIGURATION USE COUNT (CURRENT STATE)**

**Test Label:** Media Video Source Configuration Use Count Validation.

**Test Case ID:** MEDIA-2-2-6

**ONVIF Core Specification Coverage:** Get media profiles, Get video source configurations, Get video source configuration

**Command Under Test:** GetProfiles, GetVideoSourceConfigurations, GetVideoSourceConfiguration

**WSDL Reference:** media.wsdl

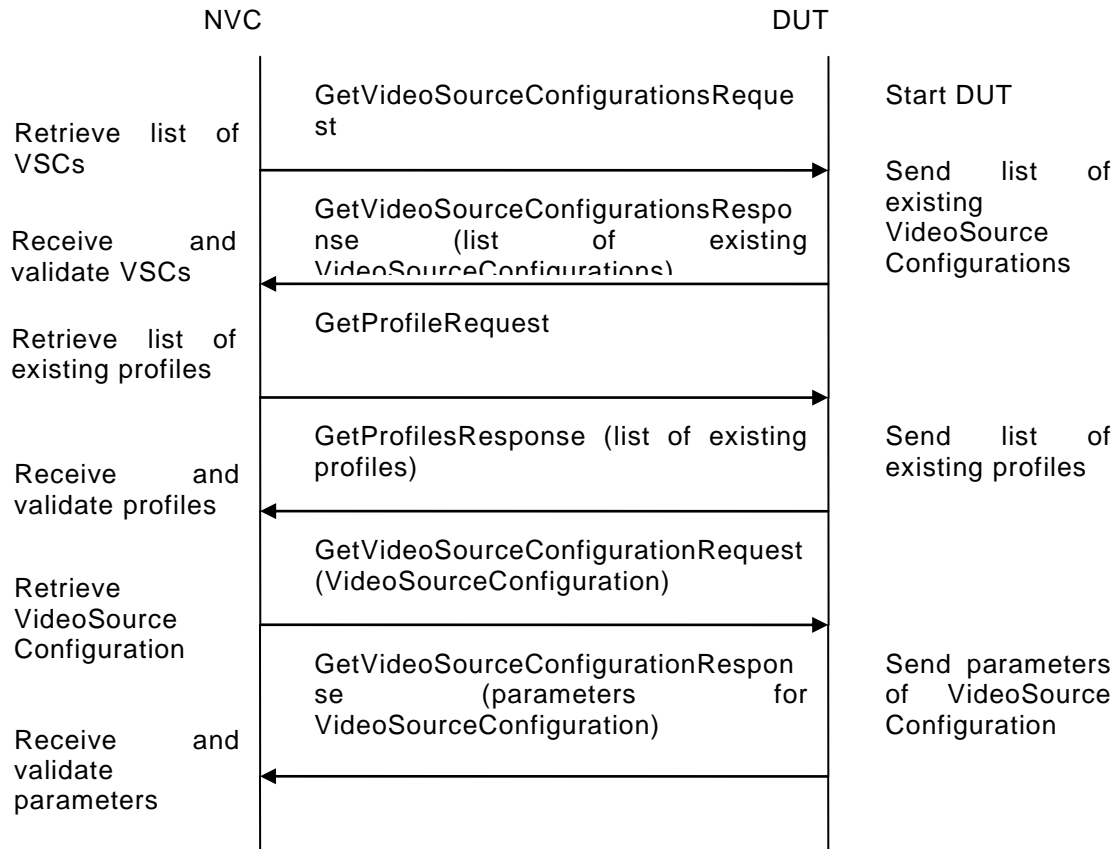
**Requirement Level:** MUST

**Test Propose:** To check Video Source Configuration use count.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

Test Configuration: NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoSourceConfigurationsRequest` message to retrieve list of video source configurations from device.
4. The DUT sends `GetVideoSourceConfigurationsResponse` message.
5. NVC invokes `GetProfilesRequest` message to retrieve list of profiles and their video source configurations from device.
6. The DUT sends `GetProfilesResponse` message.
7. Check the **UseCount=usecount1** value for the first `VideoSourceConfiguration` (VSC1) in the list on step 3.
8. Check that there are not more than **usecount1** profiles with this `VideoSourceConfiguration` in the list from step 6.
9. Check that **UseCount** value in `GetProfilesResponse` for every occurrence of this `VideoSourceConfiguration` is **usecount1**.
10. NVC invokes `GetVideoSourceConfigurationRequest` (**ConfigurationToken=the first VideoSourceConfiguration from list on step 4**) message to retrieve video source configuration parameters.

11. The DUT sends GetVideoSourceConfigurationResponse message.
12. Check the UseCount value in GetVideoSourceConfigurationResponse (Usecount=usecount1).
13. Repeat steps 7-10 for all other VideoSourceConfigurations from the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with VideoSourceConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetVideoSourceConfigurationsResponse and GetVideoSourceConfigurationResponse messages.

#### 4.3.7 VIDEO SOURCE CONFIGURATION USE COUNT (ADD SAME VIDEO SOURCE CONFIGURATION TO PROFILE TWICE)

**Test Label:** Media Service DUT Video Source Configuration Use Count Validation after Adding of Same Video Source Configuration to Profile Twice.

**Test Case ID:** MEDIA-2-2-7

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile, Add video source configuration to a profile, Get video source configurations, Get video source configuration.

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSourceConfiguration, AddVideoSourceConfiguration

**WSDL Reference:** media.wsdl

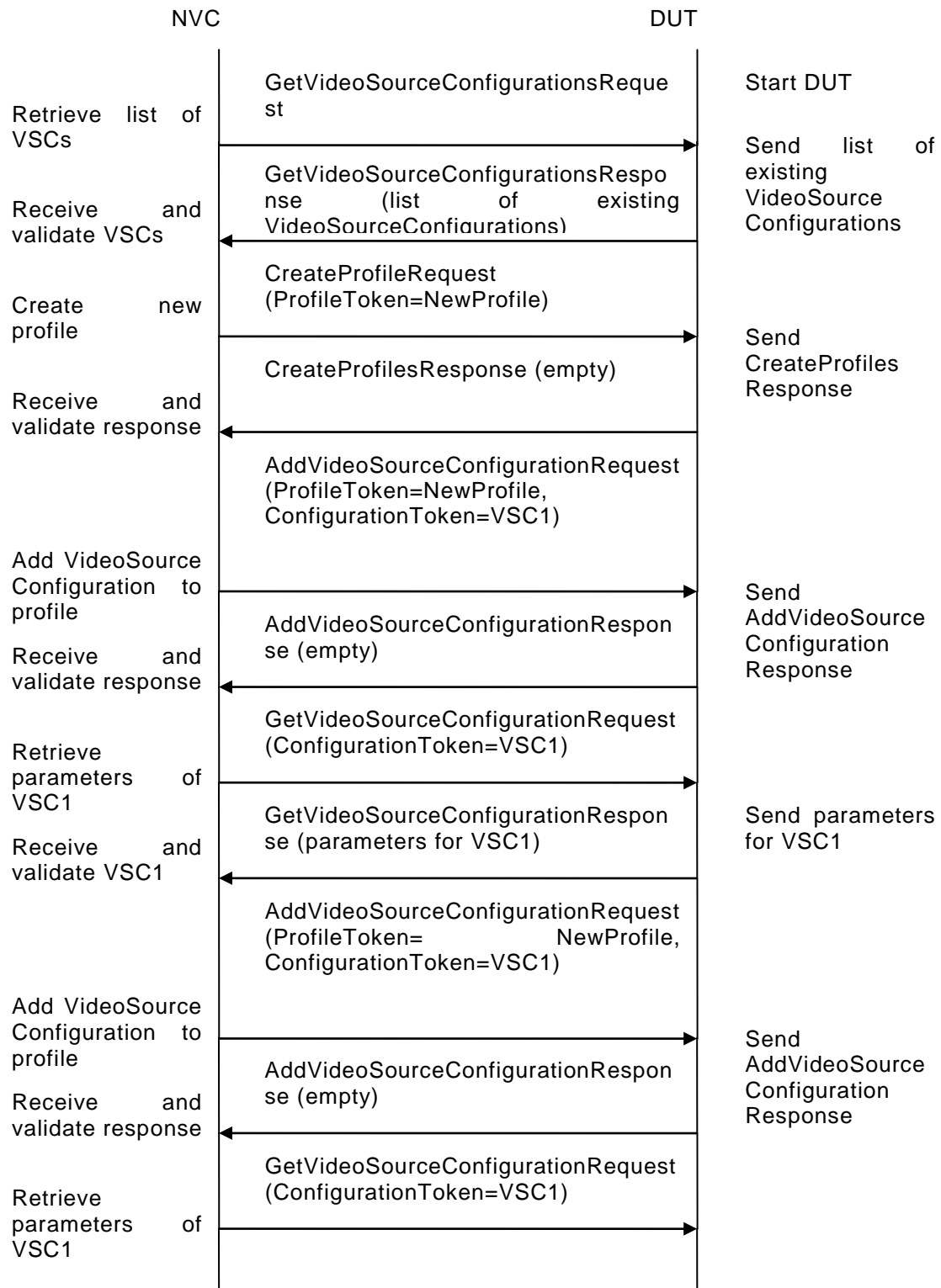
**Requirement Level:** MUST

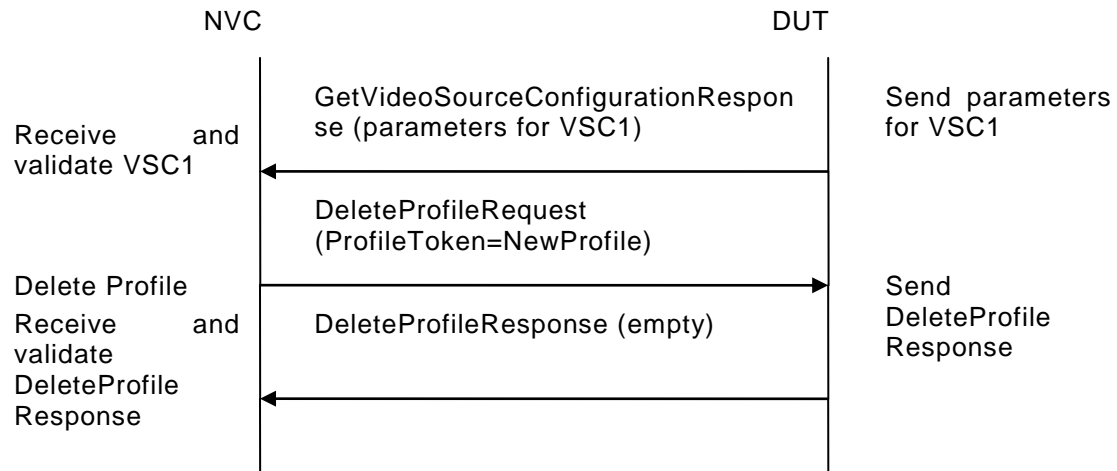
**Test Propose:** To check Video Source Configuration use count after adding same video source configuration to profile twice.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

Test Sequence:





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
4. The DUT sends GetVideoSourceConfigurationsResponse message.
5. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
6. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),

If there is profile with “fixed” attribute value “false” in profiles list (**profile1**)

NVC invokes RemoveVideoEncoderConfigurationRequest and RemoveVideoSourceConfigurationRequest (**ProfileToken=profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount of value of the VideoSourceConfiguration is reduced by 1.

Execute steps 8-17

If there is no profile with “fixed” attribute value “false” in profiles list, end test.

7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddVideoSourceConfigurationRequest (**ConfigurationToken=first video source from the list on step 4 (VSC1), ProfileToken=NewProfile**) message to add VideoSourceConfiguration to the new profile.
9. The DUT sends AddVideoSourceConfigurationResponse message.
10. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters.

11. The DUT sends GetVideoSourceConfigurationResponse message.
12. Check the UseCount value in GetVideoSourceConfigurationResponse message (**UseCount=usecount1+1**, usecount1 is value of UseCount for VSC1 from the list on step 4). If step 6 is executed, the usecount1 could be reduced by 1.
13. NVC invokes AddVideoSourceConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=VSC1**) message to replace video source configuration in profile.
14. The DUT sends AddVideoSourceConfigurationResponse message.
15. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters.
16. The DUT sends GetVideoSourceConfigurationResponse message.
17. Check that **UseCount=usecount1+1**, in GetVideoSourceConfigurationResponse message.
18. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with video source configuration.
19. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoSourceConfiguration to one more profile.

UseCount value is changed 1 after repeated adding of the VideoSourceConfiguration in profile.

#### 4.3.8 VIDEO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO SOURCE CONFIGURATIONS IN PROFILE)

**Test Label:** Media Service DUT Video Source Configuration Use Count Validation after Adding Different Video Source Configurations to the Profile.

**Test Case ID:** MEDIA-2-2-8

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile, Add video source configuration to a profile, Get video source configurations, Get video source configuration.

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSourceConfiguration, AddVideoSourceConfiguration

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST

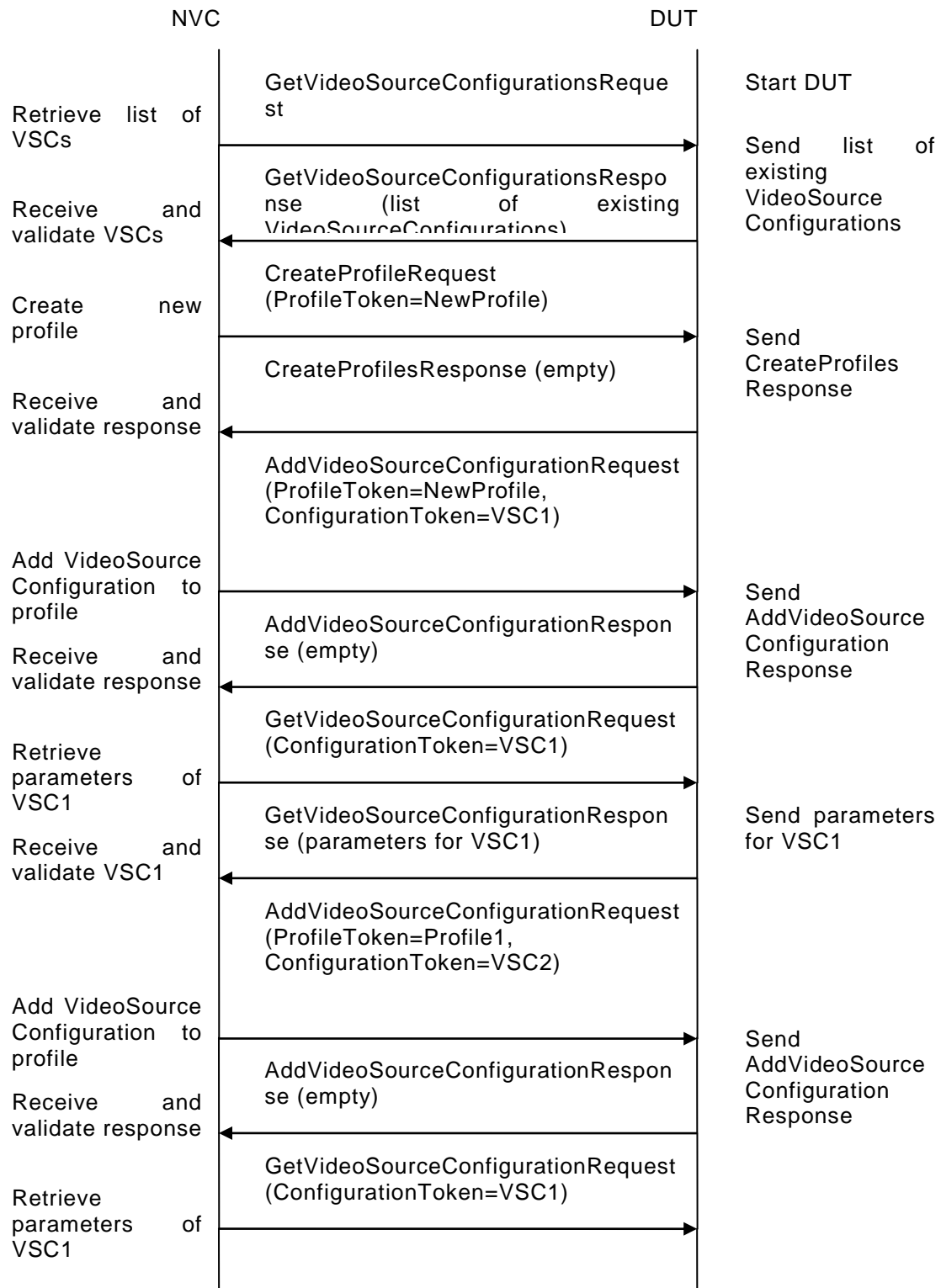
**Test Propose:** To check Video Source Configuration use count after adding different video source configurations to profile.

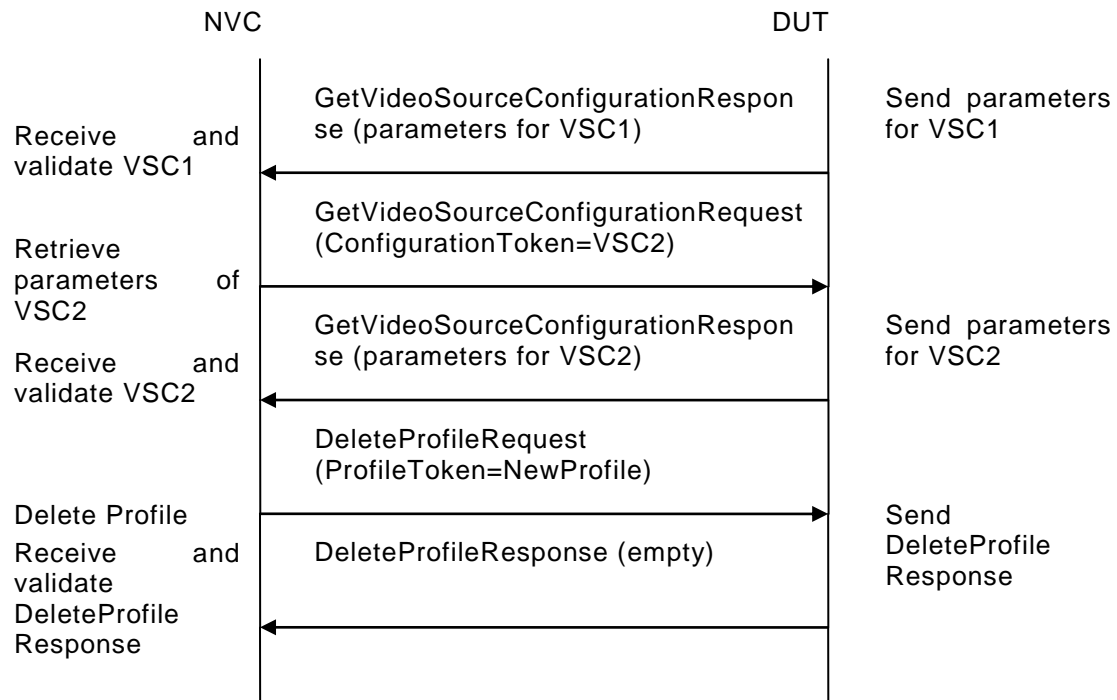
**Pre-Requisite:** Media is supported by DUT. There are at least two VideoSourceConfigurations. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
4. The DUT sends GetVideoSourceConfigurationsResponse message.
5. If there is only one VideoSourceConfiguration in the GetVideoSourceConfigurationsResponse message go to the next test.
6. NVC invokes CreateProfileRequest (**Name=NewName**, **Token=NewProfile**) message to create profile.

If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),

If there is profile with “fixed” attribute value “false” in profiles list (**profile1**)

NVC invokes RemoveVideoEncoderConfigurationRequest and RemoveVideoSourceConfigurationRequest (**ProfileToken=profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1.

The DUT sends RemoveVideoEncoderConfiguratoinResponse and RemoveVideoSourceConfigurationResponse message. The UseCount value of the two configurations will be reduced by 1.

Execute steps 8-17

If there is no profile with “fixed” attribute value “false” in profiles list, end test.

7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddVideoSourceConfigurationRequest (**ConfigurationToken=first video source from the list on step 4 (VSC1), ProfileToken=NewProfile**) message to add VideoSourceConfiguration to the new profile.
9. The DUT sends AddVideoSourceConfigurationResponse message.
10. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters.
11. The DUT sends GetVideoSourceConfigurationResponse message.
12. Check the UseCount value in GetVideoSourceConfigurationResponse message. (**UseCount=usecount1+1**, usecount1 is the value of UseCount for **VSC1** from the list on step 4). If test step 6 is executed, the usecount1 could be reduced by 1.
13. NVC invokes AddVideoSourceConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=VSC2**, where VSC2 is other VSC from list on step 4) message to replace video source configuration in profile.
14. The DUT sends AddVideoSourceConfigurationResponse message.
15. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters.
16. The DUT sends GetVideoSourceConfigurationResponse message.
17. Check that **UseCount=usecount1**, in GetVideoSourceConfigurationResponse.
18. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC2**) message to retrieve video source configuration parameters.
19. The DUT sends GetVideoSourceConfigurationResponse message.
20. Check that **UseCount=usecount2+1**, in GetVideoSourceConfigurationResponse where usecount2 is UseCount value for VSC2 in the list on step 4.
21. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with video source configuration.
22. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoSourceConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the VideoSourceConfiguration in profile for removed VideoSourceConfiguration and UseCount value is not increased by 1 after replacing of the VideoSourceConfiguration in profile for new VideoSourceConfiguration.

#### 4.3.9 VIDEO SOURCE CONFIGURATION USE COUNT (REMOVE VIDEO SOURCE CONFIGURATION)

**Test Label:** Media Service DUT Video Source Configuration Use Count Validation after Removing of Video Source Configuration in Profile Validation.

**Test Case ID:** MEDIA-2-2-9

**ONVIF Core Specification Coverage:** Remove video source configuration from a profile, Get video source configurations, Get video source configuration

**Command Under Test:** GetProfiles, GetVideoSourceConfigurations, GetVideoSourceConfiguration, RemoveVideoSourceConfiguration

**WSDL Reference:** media.wsdl

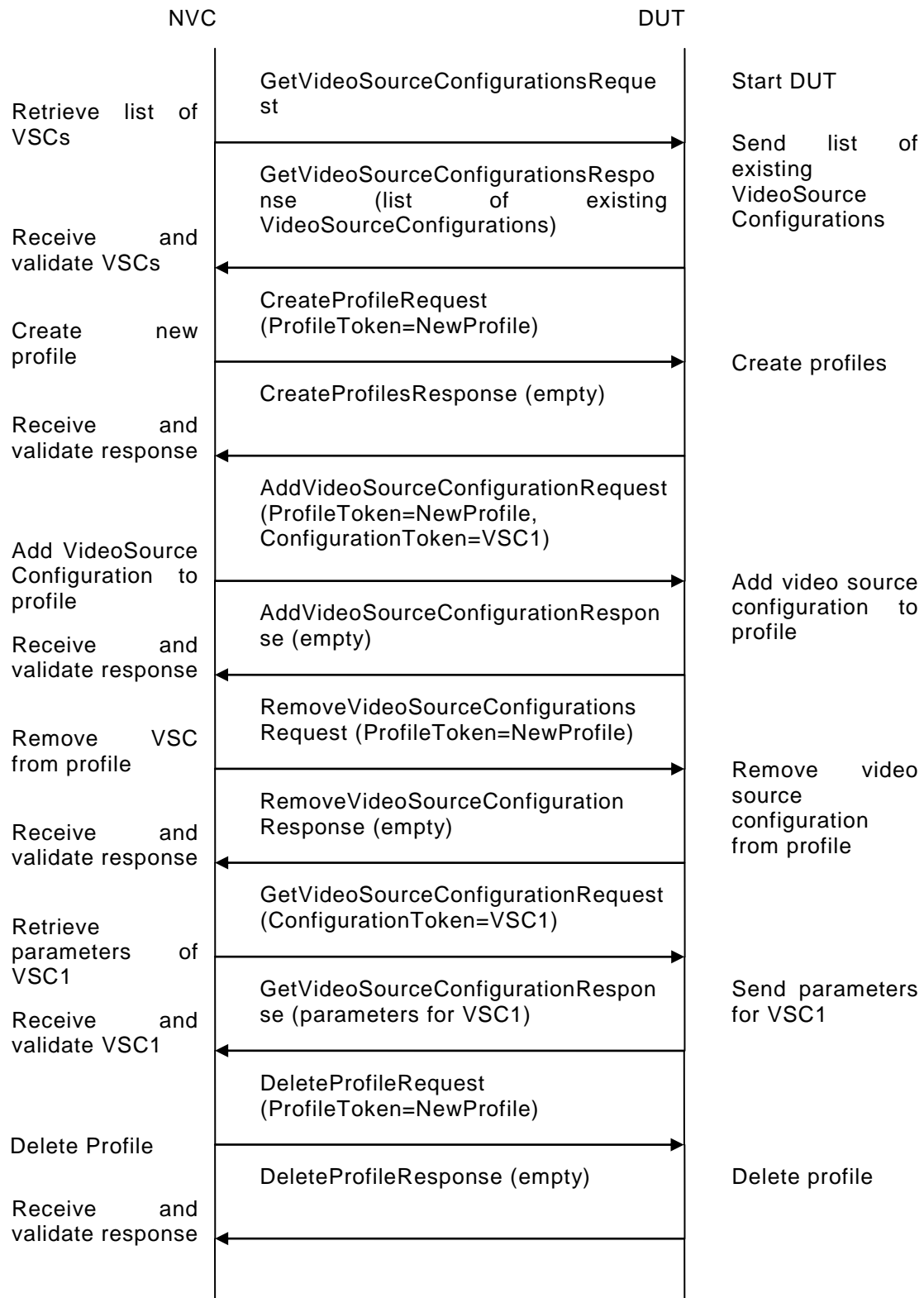
**Requirement Level:** MUST

**Test Propose:** To check Video Source Configuration use count after removing video source configuration to profile.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of media profile is received by GetProfiles command.

**Test Configuration:** NVC and DUT

Test Sequence:



Test Procedure:

1. Start an NVC.

2. Start a DUT.
3. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
4. The DUT sends GetVideoSourceConfigurationsResponse message.
5. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
6. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
 If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
 NVC invokes RemoveVideoEncoderConfigurationRequest and RemoveVideoSourceConfigurationRequest (**ProfileToken=profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1  
 The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount value of the two configurations will be reduced by 1.  
 Execute steps 8-13  
 If there is no profile with "fixed" attribute value "false" in profiles list, end test.
7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddVideoSourceConfigurationRequest (**ConfigurationToken=first video source from the list on step 4 (VSC1), ProfileToken=NewProfile**) message to add VideoSourceConfiguration to the new profile.
9. The DUT sends AddVideoSourceConfigurationResponse message.
10. NVC invokes RemoveVideoSourceConfigurationRequest (**ProfileToken=NewProfile**) message to remove video source configuration from NewProfileToken.
11. The DUT sends RemoveVideoSourceConfigurationResponse message.
12. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters. The DUT sends GetVideoSourceConfigurationResponse message.
13. Check the UseCount value in GetVideoSourceConfigurationResponse. (**UseCount=usecount1**, usecount1 is the UseCount value for **VSC1** in the list on step 4). If test step 6 is executed, the usecount1 value could be reduced by 1.
14. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile.
15. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.

The DUT did not send valid RemoveVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not decreased by 1 after removing of the VideoSourceConfiguration from profile.

#### **4.3.10 VIDEO SOURCE CONFIGURATION USE COUNT (DELETION PROFILE WITH VIDEO SOURCE CONFIGURATION)**

**Test Label:** Media Service DUT Video Source Configuration Use Count Validation after Deletion of Profile with Video Source Configuration Validation.

**Test Case ID:** MEDIA-2-2-10

**ONVIF Core Specification Coverage:** Get video source configurations, Get video source configuration, Delete media profile

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSourceConfiguration, DeleteProfile.

**WSDL Reference:** media.wsdl

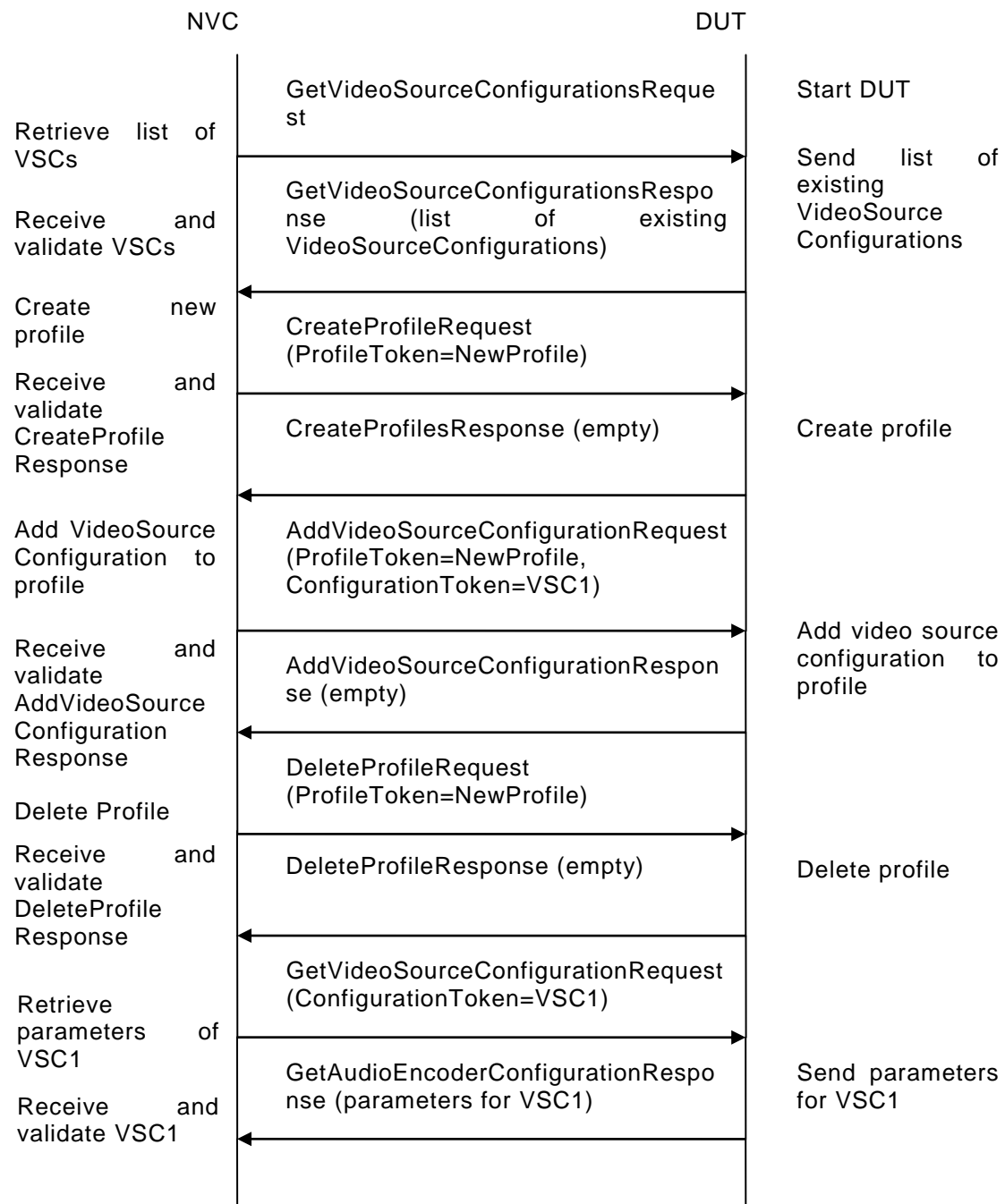
**Requirement Level:** MUST

**Test Propose:** To check Video Source Configuration use count after deletion of profile with source configuration in it.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles.

**Test Configuration:** NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
4. The DUT sends GetVideoSourceConfigurationsResponse message.



5. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
6. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
NVC invokes RemoveVideoEncoderConfiguration and RemoveVideoSourceConfigurationRequest (**ProfileToken=profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1  
The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount values for the two configurations will be reduced by 1.  
Execute steps 8-14  
If there is no profile with "fixed" attribute value "false" in profiles list, end test.
7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddVideoSourceConfigurationRequest (**ConfigurationToken=first video source from the list on step 4 (VSC1), ProfileToken=NewProfile**) message to add VideoSourceConfiguration to the new profile.
9. The DUT sends AddVideoSourceConfigurationResponse message.
10. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with video source configuration.
11. The DUT sends DeleteProfileResponse message.
12. NVC invokes GetVideoSourceConfigurationRequest (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters.
13. The DUT sends GetVideoSourceConfigurationResponse message.
14. Check the UseCount value in GetVideoSourceConfigurationResponse message. (**UseCount=usecount1**, usecount1 is the UseCount value for VSC1 in the list on step 4). If test step 6 is executed, the usecount value could be reduced by 1.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.

The DUT did not send valid RemoveVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the VideoSourceConfiguration.

#### 4.3.11 VIDEO SOURCE CONFIGURATION USE COUNT (SET VIDEO SOURCE CONFIGURATION)

**Test Label:** Media Service DUT Video Source Configuration Use Count Validation after Changinf of Video Source Configuration Validation.

**Test Case ID:** MEDIA-2-2-11

**ONVIF Core Specification Coverage:** Get video source configurations, Get video source configuration, Modify a video source configuration.

**Command Under Test:** GetVideoSourceConfigurations, GetVideoSourceConfiguration, SetVideoSourceConfiguration

**WSDL Reference:** media.wsdl

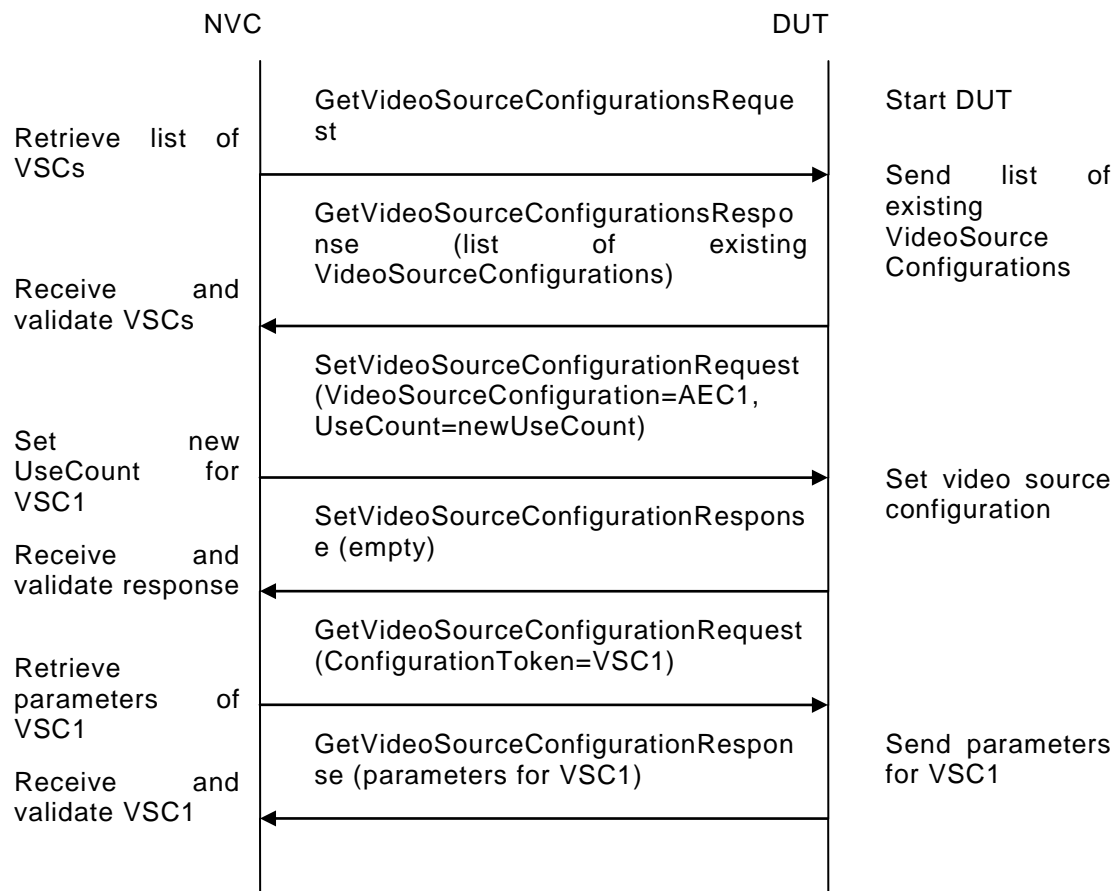
**Requirement Level:** MUST

**Test Propose:** To check Video Source Configuration use count after setting video source configuration parameters.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoSourceConfigurationsRequest` message to retrieve list of video source configurations from device.
4. The DUT sends `GetVideoSourceConfigurationsResponse` message.
5. NVC invokes `SetVideoSourceConfigurationRequest` (**token=the first Video Source Configuration from the list on step 4(VSC1), UseCount=NewUseCount**) message to set parameters for video source interval profile with video source configuration.
6. The DUT sends `SetVideoSourceConfigurationResponse` message.
7. NVC invokes `GetVideoSourceConfigurationRequest` (**ConfigurationToken=VSC1**) message to retrieve video source configuration parameters.
8. The DUT sends `GetVideoSourceConfigurationResponse` message.
9. Check that **UseCount =usecount1**, in `GetVideoSourceConfigurationResponse` where usecount1 is UseCount value for VSC1 in the list on step 4.

#### Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send SetVideoSourceConfigurationResponse message.

The DUT did not send valid SetVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value changed after trying to set UseCount value.

## 4.4 Video Encoder Configuration

### 4.4.1 VIDEO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY

**Test Label:** Media Service DUT GetVideoEncoderConfigurations Command and GetProfiles Command Consistency Validation.

**Test Case ID:** MEDIA-2-3-1

**ONVIF Core Specification Coverage:** Get media profiles, Get video encoder configurations

**Command Under Test:** GetProfiles, GetVideoEncoderConfigurations

WSDL Reference: media.wsdl

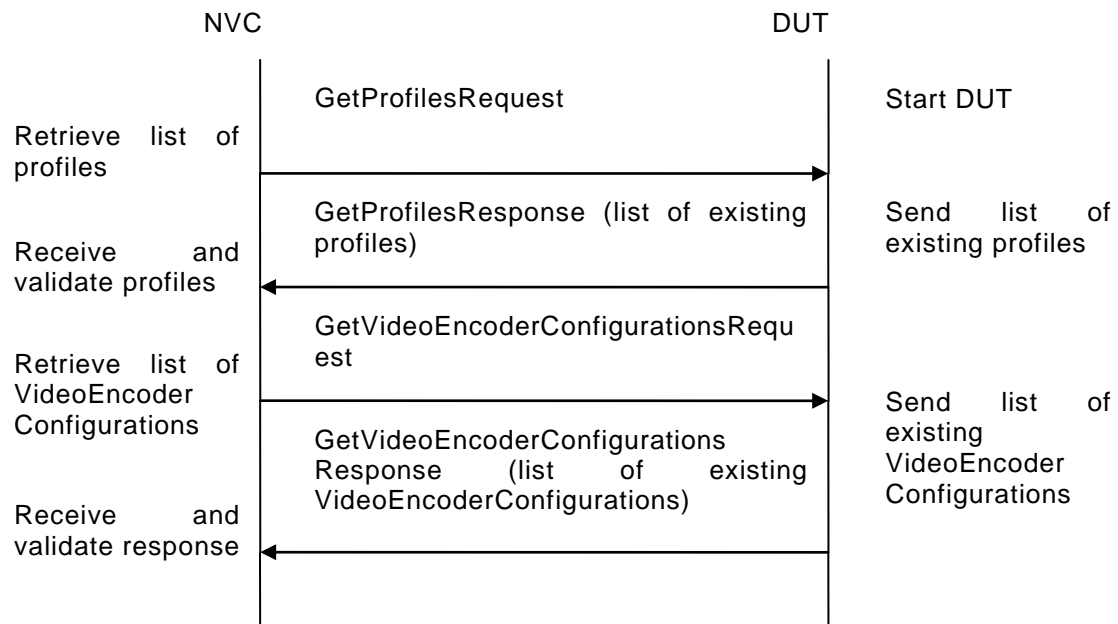
**Requirement Level:** MUST

**Test Propose:** To check that GetVideoEncoderConfigurations command and GetProfiles command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

Test Sequence:



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetProfilesRequest message to retrieve list of profiles and their video encoder configurations.
4. The DUT sends GetProfilesResponse message.
5. NVC invokes GetVideoEncoderConfigurationsRequest message to retrieve list of Video Encoder Configurations from device.
6. The DUT sends GetVideoEncoderConfigurationsResponse message.
7. Check that each VideoEncoderConfiguration from GetVideoEncoderConfigurationsResponse message has unique token.
8. Check that all VideoEncoderConfigurations from the GetProfilesResponse message are included in the GetVideoEncoderConfigurationsResponse message.
9. Check that VideoEncoderConfiguration parameters are same in GetProfilesResponse message and in GetVideoEncoderConfigurationsResponse message for each VideoEncoderConfiguration.

#### Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT return two or more VideoEncoderConfigurations in GetVideoEncoderConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with VideoEncoderConfigurations that were not included in the GetVideoEncoderConfigurationsResponse message.

The DUT returned different parameters list and parameters values for the same VideoEncoderConfiguration in the GetVideoEncoderConfigurationsResponse message and in the GetProfilesResponse message.

#### 4.4.2 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION

##### CONSISTENCY

**Test Label:** Media Service DUT GetVideoEncoderConfigurations Command and GetVideoEncoderConfiguration Command Consistency Validation.

**Test Case ID:** MEDIA-2-3-2

**ONVIF Core Specification Coverage:** Get media profiles, Get video encoder configurations

**Command Under Test:** GetVideoEncoderConfiguration, GetVideoEncoderConfigurations

**WSDL Reference:** media.wsdl

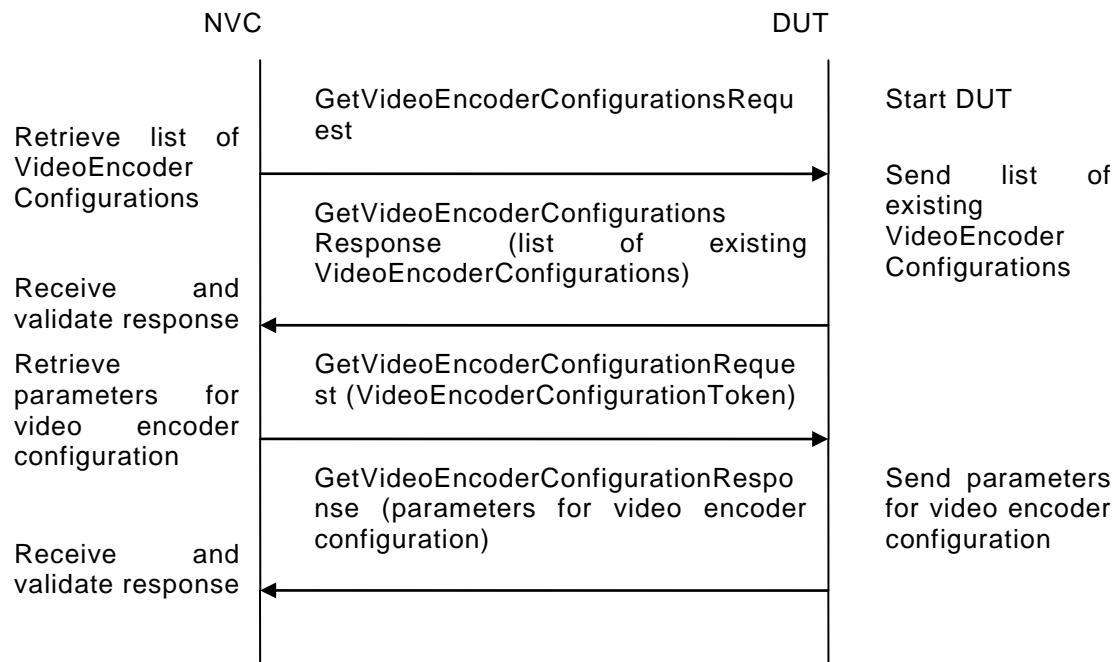
**Requirement Level:** MUST

**Test Propose:** To check that GetVideoEncoderConfigurations command and GetVideoEncoderConfiguration command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoEncoderConfigurationsRequest message to retrieve list of Video Encoder Configurations from device.
4. The DUT sends GetVideoEncoderConfigurationsResponse message.
5. NVC invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken**) message to retrieve parameters of video encoder configuration from device.
6. The DUT sends GetVideoEncoderConfigurationResponse message.
7. Check that all parameters for video encoder configuration in the GetVideoEncoderConfigurationsResponse message and GetVideoEncoderConfigurationResponse message are the same.
8. Repeat steps 5-7 for each Video Encoder Configurations from the GetVideoEncoderConfigurationsResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT did not send equal parameters for one or more VideoEncoderConfiguration in the GetVideoEncoderConfigurationResponse message and in the GetVideoEncoderConfigurationsResponse message.

#### 4.4.3 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION OPTIONS

##### CONSISTENCY

**Test Label:** Media Service DUT GetVideoEncoderConfigurations Command and GetVideoEncoderConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-2-3-3

**ONVIF Core Specification Coverage:** Get video encoder configurations, Get video encoder configuration options

**Command Under Test:** GetVideoEncoderConfigurations, GetVideoEncoderConfigurationOptions

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST

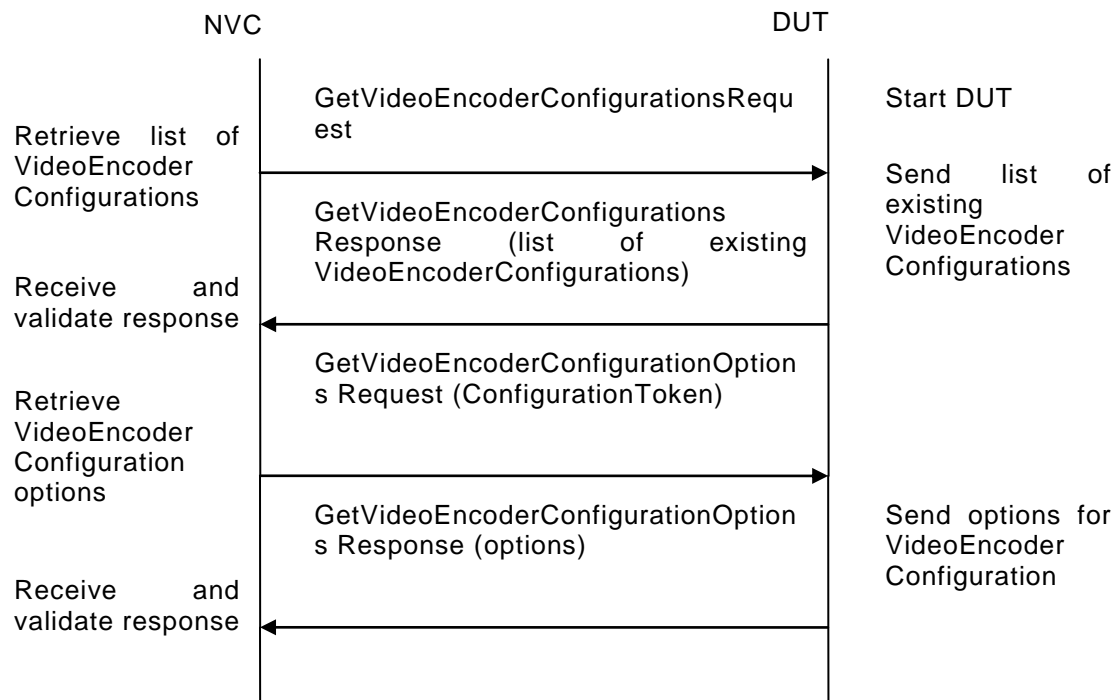
**Test Propose:** To check that GetVideoEncoderConfigurations command and GetVideoEncoderConfigurationOptions command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` message to retrieve list of Video Encoder Configurations from device.
4. The DUT sends `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoEncoderConfigurationOptionsRequest (ConfigurationToken)` message to retrieve `VideoEncoderConfigurationOptions` from device.
6. The DUT sends `GetVideoEncoderConfigurationOptionsResponse` message.
7. Check that all parameters of video encoder configuration are available according to options for video encoder configuration.
8. Repeat steps 5-7 for other video encoder configurations from the `GetVideoEncoderConfigurationsResponse` message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send `GetVideoEncoderConfigurationsResponse` message.

The DUT did not send valid `GetVideoEncoderConfigurationsResponse` message.

The DUT did not send GetVideoEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationOptionsResponse message.

The DUT sent for at list one VideoEncoderConfiguration with inconsistent parameters:

- Encoding value from video encoder configuration is not included in options
- There is no pair "Resolution.High, Resolution.Width" from video encoder configuration in AvailableResolution list for previously checked Encoder from options.
- Quality value for video encoder configuration is not between QualityRange.Min and QualityRange.Max from options.
- RateControl.FrameRateLimit for video encoder configuration is not between Encoding.FrameRateRange.Min and Encoding.FrameRateRange.Max for current encoding from options
- RateControl.EncodingInterval for video encoder configuration is not between EncodingIntervalRange.Min and EncodingIntervalRange.Max for current encoding from options.
- RateControl.BitrateLimit for video encoder configuration is not between BitrateRange.Min and BitrateRange.Max for current encoding from options.
- If Encoding is H.264, H264.H264Profile value does not exist in H264.H264ProfilesSupported list from options.
- If Encoding is MPEG4, MPEG4.MPEG4Profile value does not exist in MPEG4.MPEG4ProfilesSupported list from options.
- If Encoding is H264 or MPEG4, Enc.GovLength value is not between Enc.GovLengthRange.Min and Enc.GovLengthRange.Max for corresponded encoding from options.

#### 4.4.4 PROFILES AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY

**Test Label:** Media Service DUT GetProfiles Command and GetVideoEncoderConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-2-3-4

**ONVIF Core Specification Coverage:** Get media profile, Get video encoder configuration options

**Command Under Test:** GetProfiles, GetVideoEncoderConfigurationOptions

**WSDL Reference:** media.wsdl

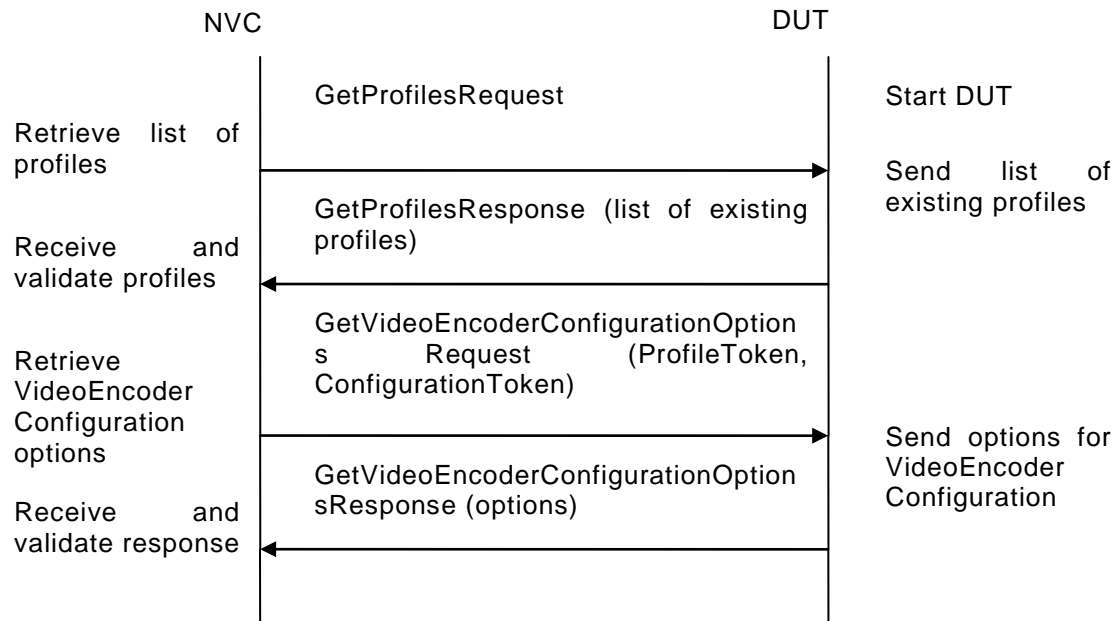
**Requirement Level:** MUST

**Test Propose:** To check that GetProfiles command and GetVideoEncoderConfigurationOptions command are consistent.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes **GetProfilesRequest** message to retrieve list of profiles from device.
4. The DUT sends **GetProfilesResponse** message.
5. NVC invokes **GetVideoEncoderConfigurationOptionsRequest** (**ProfileToken**, **ConfigurationToken**) message to retrieve **VideoEncoderConfigurationOptions** for video encoder configuration from the DUT.
6. The DUT sends **GetVideoEncoderConfigurationOptionsResponse** message.
7. Check that all parameters of video encoder configuration in profile are available according to options for video encoder configuration.
8. Repeat steps 5 -7 for other profiles from the **GetProfilesResponse** message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send **GetProfilesResponse** message.

The DUT did not send valid **GetProfilesResponse** message.

The DUT did not send **GetVideoEncoderConfigurationOptionsResponse** message.

The DUT did not send valid **GetVideoEncoderConfigurationOptionsResponse** message.

The DUT sent for at list one VideoEncoderConfiguration with inconsistent parameters:

- Encoding value from video encoder configuration is not included in options
- There is no pair “Resolution.High, Resolution.Width” from video encoder configuration in AvailableResolution list for previously checked Encoder from options.
- Quality value for video encoder configuration is not between QualityRange.Min and QualityRange.Max from options.
- RateControl.FrameRateLimit for video encoder configuration is not between Encoding.FrameRateRange.Min and Encoding.FrameRateRange.Max for current encoding from options
- RateControl.EncodingInterval for video encoder configuration is not between EncodingIntervalRange.Min and EncodingIntervalRange.Max for current encoding from options.
- RateControl.BitrateLimit for video encoder configuration is not between BitrateRange.min and BitrateRange.max for current encoding from options.
- If Encoding is H.264, H264.H264Profile value does not exist in H264.H264ProfilesSupported list from options.
- If Encoding is MPEG4, MPEG4.MPEG4Profile value does not exist in MPEG4.MPEG4ProfilesSupported list from options.
- If Encoding is H264 or MPEG4, Enc.GovLength value is not between Enc.GovLengthRange.Min and Enc.GovLengthRange.Max for corresponded encoding from options.

#### 4.4.5 VIDEO ENCODER CONFIGURATION USE COUNT (CURRENT STATE)

**Test Label:** Media Service DUT Video Encoder Configuration Use Count Validation.

**Test Case ID:** MEDIA-2-3-5

**ONVIF Core Specification Coverage:** Get media profiles, Get video encoder configurations, Get video encoder configuration

**Command Under Test:** GetProfiles, GetVideoEncoderConfigurations, GetVideoEncoderConfiguration

**WSDL Reference:** media.wsdl

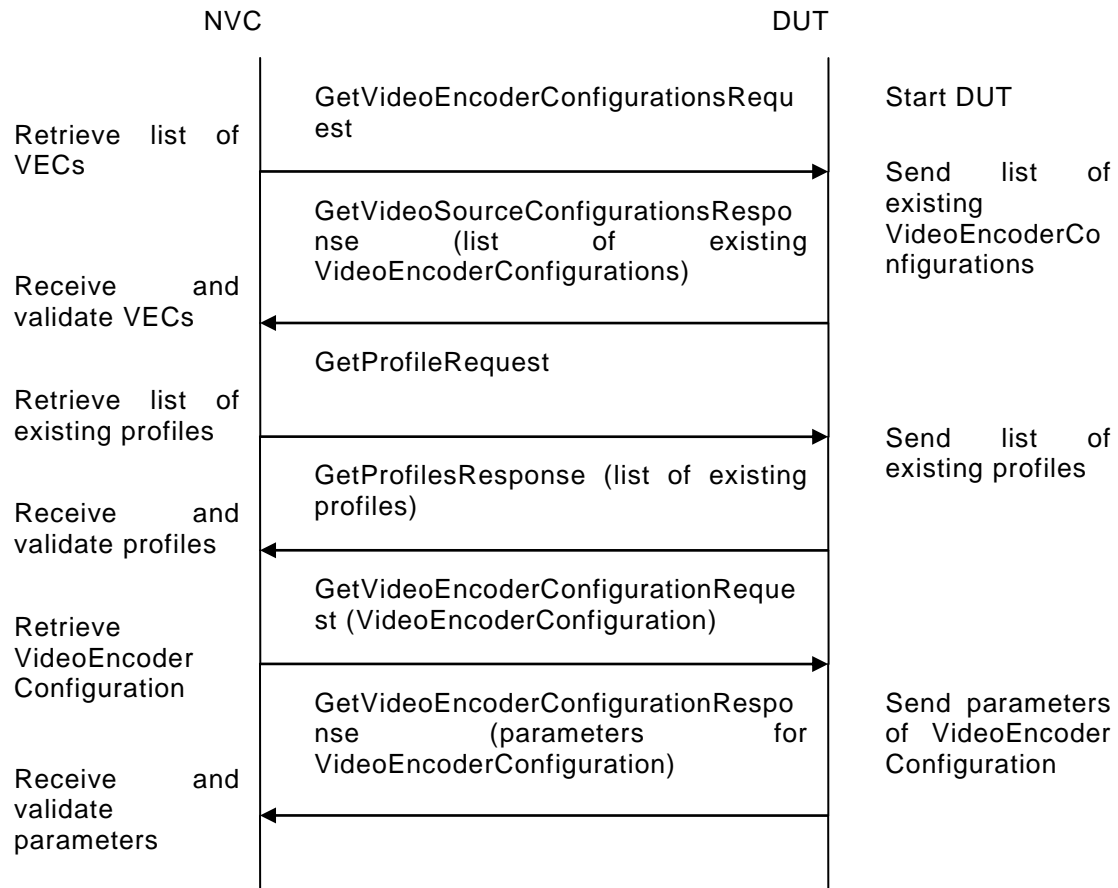
**Requirement Level:** MUST

**Test Propose:** To check Video Encoder Configuration use count.

**Pre-Requisite:** Media is supported by DUT. Profile with VideoEncoderConfiguration exists. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` message to retrieve list of video encoder configurations from device.
4. The DUT sends `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetProfilesRequest` message to retrieve list of profiles and their video encoder configurations from device.
6. The DUT sends `GetProfilesResponse` message.
7. Check the **UseCount=usecount1** value for the first `VideoEncoderConfiguration` (VEC1) in the list on step 4.
8. Check that there are not more than **usecount1** profiles with this `VideoEncoderConfiguration` in the list from step 6.
9. Check that **UseCount** value in `GetProfilesResponse` for every occurrence of this `VideoEncoderConfiguration` is **usecount1**.
10. NVC invokes `GetVideoEncoderConfigurationRequest` (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.

11. The DUT sends GetVideoEncoderConfigurationResponse message.
12. Check the UseCount value in GetVideoEncoderConfigurationResponse (Usecount=usecount1).
13. Repeat steps 7-10 for all other VideoEncoderConfigurations from the list on step 4.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with VideoEncoderConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetVideoEncoderConfigurationsResponse and GetVideoEncoderConfigurationResponse messages.

#### 4.4.6 VIDEO ENCODER CONFIGURATION USE COUNT (ADD SAME VIDEO ENCODER CONFIGURATION TO PROFILE TWICE)

**Test Label:** Media Service DUT Video Encoder Configuration Use Count Validation after Adding of Same Video Encoder Configuration to Profile Twice.

**Test Case ID:** MEDIA-2-3-6

**ONVIF Core Specification Coverage:** Add video encoder configuration to a profile, Get video encoder configurations, Get video encoder configuration.

**Command Under Test:** GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, AddVideoEncoderConfiguration

**WSDL Reference:** media.wsdl

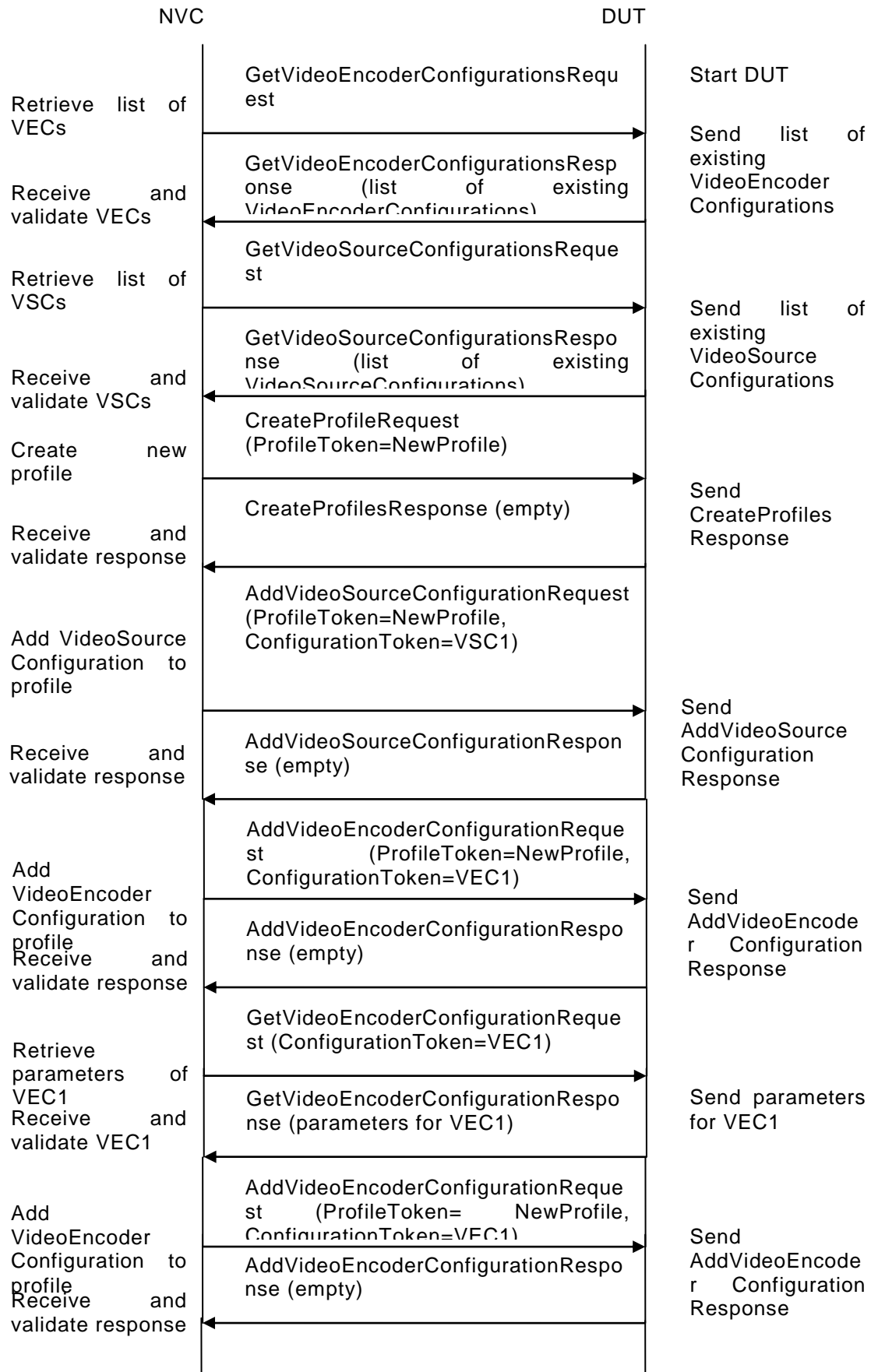
**Requirement Level:** MUST

**Test Propose:** To check Video Encoder Configuration use count after adding same video encoder configuration to profile twice.

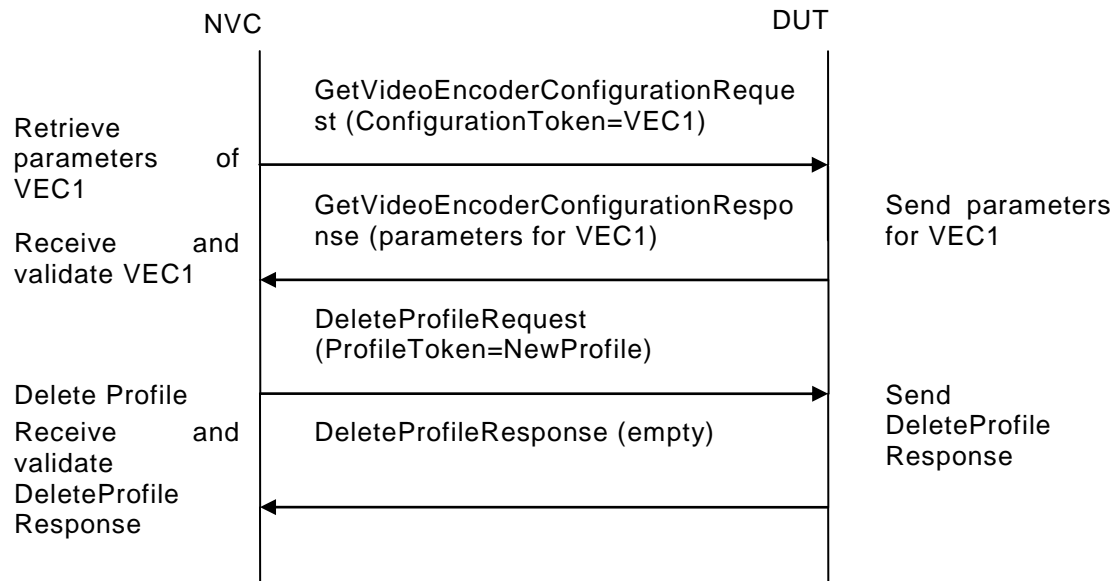
**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` message to retrieve list of video encoder configurations from device.
4. The DUT sends `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoSourceConfigurationsRequest` message to retrieve list of video source configurations from device.
6. The DUT sends `GetVideoSourceConfigurationsResponse` message.
7. NVC invokes `CreateProfileRequest` (**Name=NewName, Token=NewProfile**) message to create profile.
8. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
NVC invokes `RemoveVideoEncoderConfigurationRequest` (**ProfileToken=profile1**) message to remove `VideoEncoderConfiguration` from profile1  
The DUT sends `RemoveVideoEncoderConfigurationResponse` message  
Execute steps 10-21  
If there is no profile with "fixed" attribute value "false" in profiles list, end test.
9. If the DUT sends `CreateProfileResponse` message, validate `CreateProfileResponse` message from the DUT.

10. NVC invokes AddVideoSourceConfigurationRequest (**ConfigurationToken=first video source from the list on step 6 (VSC1), ProfileToken=NewProfile**) message to add VideoSourceConfiguration to the new profile.
11. The DUT sends AddVideoSourceConfigurationResponse message.
12. NVC invokes AddVideoEncoderConfigurationRequest (**ConfigurationToken=first video encoder from the list on step 4 (VEC1), ProfileToken=NewProfile**) message to add VideoEncoderConfiguration to the new profile.
13. The DUT sends AddVideoEncoderConfigurationResponse message.
14. NVC invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.
15. The DUT sends GetVideoEncoderConfigurationResponse message.
16. Check the UseCount value in GetVideoEncoderConfigurationResponse message (**UseCount=usecount1+1**, usecount1 is the UseCount value for VEC1 from the list on step 4). If test step 8 is executed, the usecount1 value could be reduced by 1.
17. NVC invokes AddVideoEncoderConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=VEC1**) message to replace video encoder configuration in profile.
18. The DUT sends AddVideoEncoderConfigurationResponse message.
19. NVC invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.
20. The DUT sends GetVideoEncoderConfigurationResponse message.
21. Check that **UseCount=usecount1+1**, in GetVideoEncoderConfigurationResponse message.
22. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with video encoder configuration.
23. The DUT sends DeleteProfileResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoEncoderConfiguration to one more profile.

UseCount value is changed after repeated adding of the VideoEncoderConfiguration in profile.

#### **4.4.7 VIDEO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO ENCODER CONFIGURATIONS IN PROFILE)**

**Test Label:** Media Service DUT Video Encoder Configuration Use Count Validation after Adding Different Video Encoder Configurations to the Profile.

**Test Case ID:** MEDIA-2-3-7

**ONVIF Core Specification Coverage:** Add video encoder configuration to a profile, Get video encoder configurations, Get video encoder configuration.

**Command Under Test:** GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, AddVideoEncoderConfiguration

**WSDL Reference:** media.wsdl

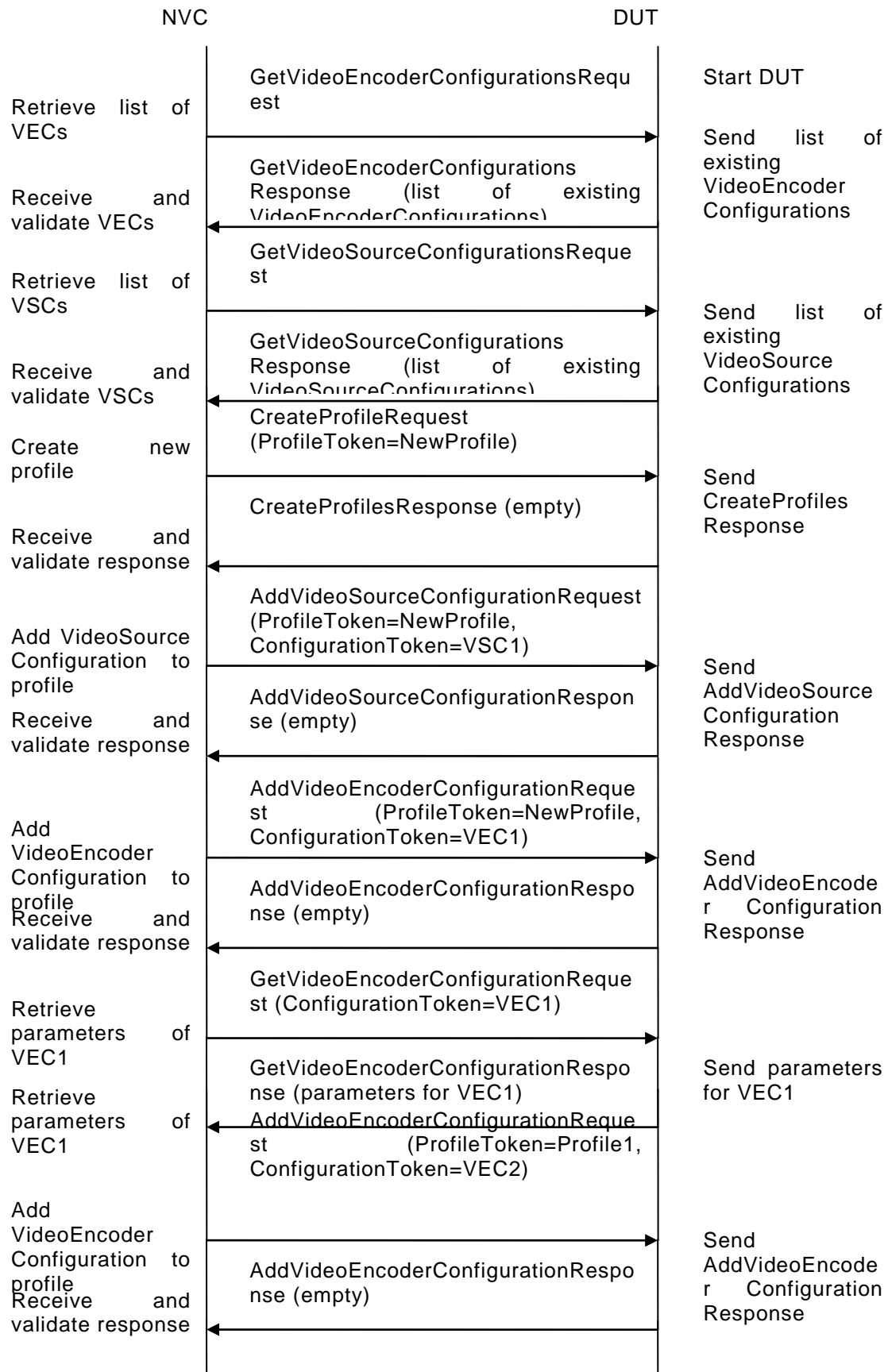
**Requirement Level:** MUST

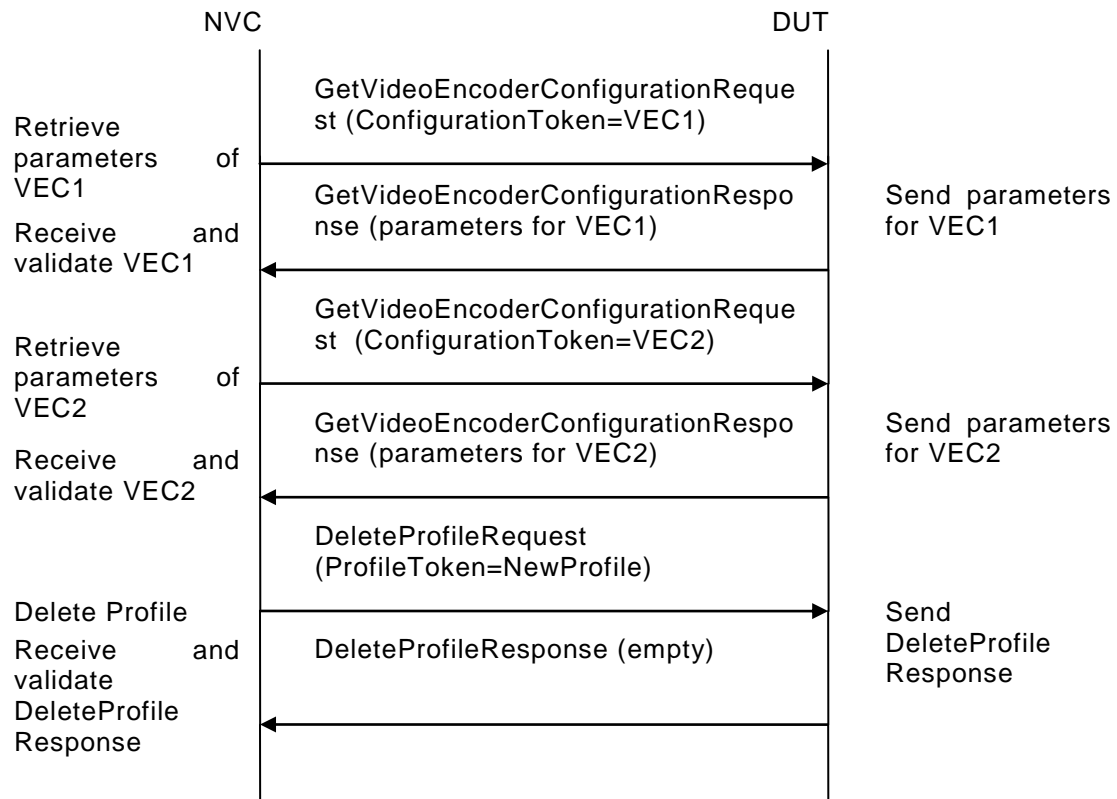
**Test Propose:** To check Video Encoder Configuration use count after adding different video encoder configurations to profile.

**Pre-Requisite:** Media is supported by DUT. There are at least two VideoEncoderConfigurations. NVC gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
4. The DUT sends GetVideoEncoderConfigurationsResponse message.
5. If there is only one VideoEncoderConfiguration in the GetVideoEncoderConfigurationsResponse message go to the next test.
6. NVC invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
7. The DUT sends GetVideoSourceConfigurationsResponse message.
8. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.

If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

NVC invokes RemoveVideoEncoderConfigurationRequest (**ProfileToken=profile1**) message to remove VideoEncoderConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse message

Execute steps 10-21

If there is no profile with “fixed” attribute value “false” in profiles list, end test.

9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
10. NVC invokes AddVideoSourceConfigurationRequest (**ConfigurationToken=first video source from the list on step 7 (VSC1), ProfileToken=NewProfile**) message to add VideoSourceConfiguration to the new profile.
11. The DUT sends AddVideoSourceConfigurationResponse message.
12. NVC invokes AddVideoEncoderConfigurationRequest (**ConfigurationToken=first video encoder from the list on step 4 (VEC1), ProfileToken=NewProfile**) message to add VideoEncoderConfiguration to the new profile.
13. The DUT sends AddVideoEncoderConfigurationResponse message.
14. NVC invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.
15. The DUT sends GetVideoEncoderConfigurationResponse message.
16. Check the UseCount value in GetVideoEncoderConfigurationResponse (**UseCount=usecount1+1**, usecount1 is value of UseCount for VEC1 from the list on step 4). If step 8 is executed, the usecount1 could be reduced by 1.
17. NVC invokes AddVideoEncoderConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=VEC2**, where VEC2 is other VEC from list on step 4) message to replace video encoder configuration in profile.
18. The DUT sends AddVideoEncoderConfigurationResponse message.
19. NVC invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.
20. The DUT sends GetVideoEncoderConfigurationResponse message.
21. Check that **UseCount=usecount1**, in GetVideoEncoderConfigurationResponse.
22. NVC invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken=VEC2**) message to retrieve video encoder configuration parameters.
23. The DUT sends GetVideoEncoderConfigurationResponse message.
24. Check that **UseCount=usecount2+1**, in GetVideoEncoderConfigurationResponse where usecount2 is UseCount value for VEC2 in the list on step 4.
25. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with video encoder configuration.
26. The DUT sends DeleteProfileResponse message.

**Test Result:**

**PASS –**

DUT passes all assertions.

#### **FAIL –**

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoEncoderConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the VideoEncoderConfiguration in profile for removed VideoEncoderConfiguration and UseCount value is not increased by 1 after replacing of the VideoEncoderConfiguration in profile for new VideoEncoderConfiguration.

#### **4.4.8 VIDEO ENCODER CONFIGURATION USE COUNT (REMOVE VIDEO ENCODER CONFIGURATION)**

**Test Label:** Media Service DUT Video Encoder Configuration Use Count Validation after Removing of Video Encoder Configuration in Profile Validation.

**Test Case ID:** MEDIA-2-3-8

**ONVIF Core Specification Coverage:** Remove video encoder configuration from a profile, Get video encoder configurations, Get video encoder configuration.

**Command Under Test:** GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, RemoveVideoEncoderConfiguration

**WSDL Reference:** media.wsdl

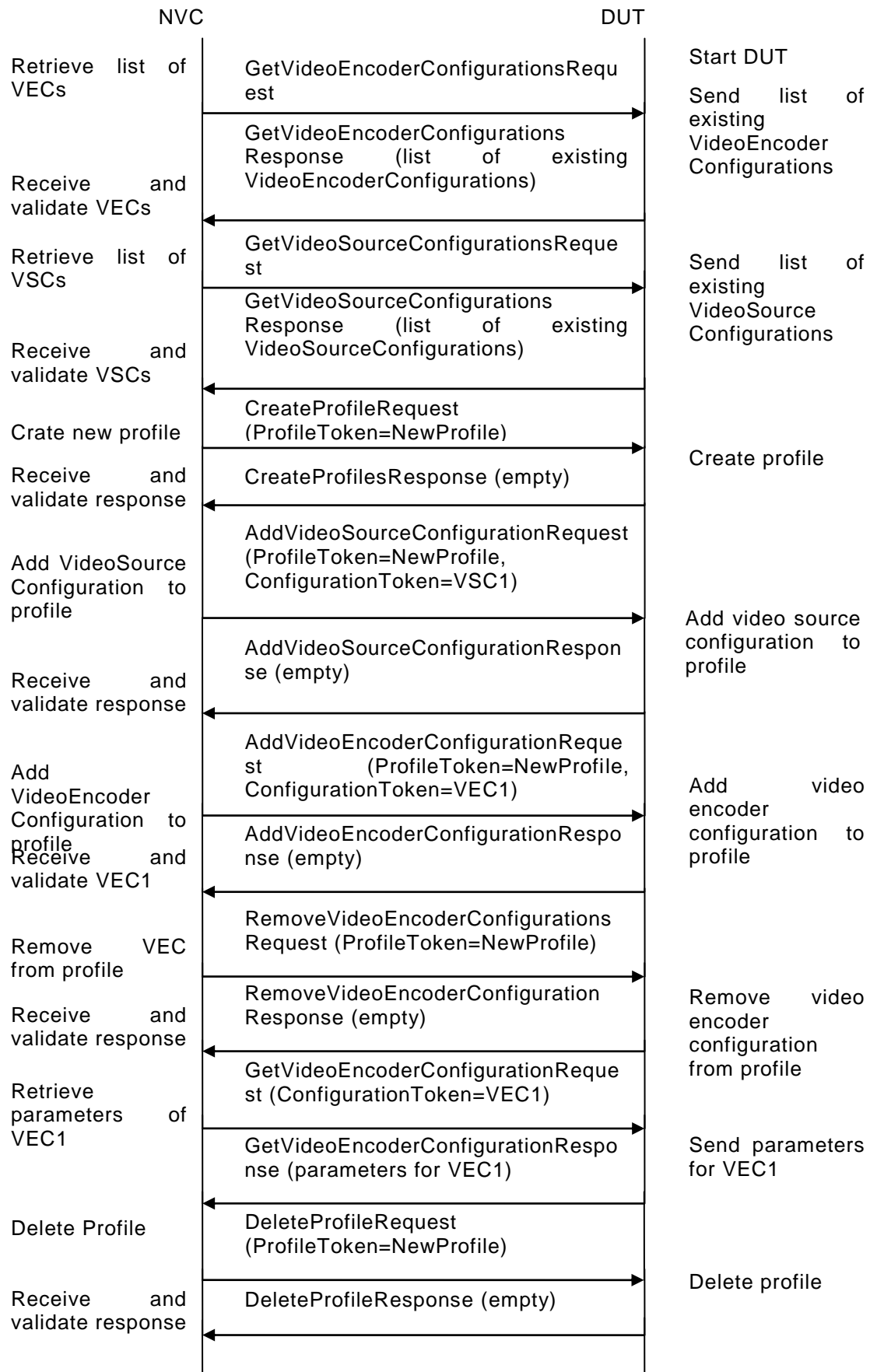
**Requirement Level:** MUST

**Test Propose:** To check Video Encoder Configuration use count after removing video encoder configuration from a profile.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

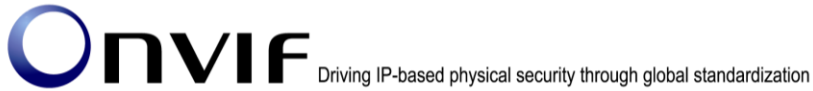
Test Sequence:





### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` message to retrieve list of video encoder configurations from device.
4. The DUT sends `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoSourceConfigurationsRequest` message to retrieve list of video source configurations from device.
6. The DUT sends `GetVideoSourceConfigurationsResponse` message.
7. NVC invokes `CreateProfileRequest` (**Name=NewName, Token=NewProfile**) message to create profile.
8. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
 If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
 NVC invokes `RemoveVideoEncoderConfigurationRequest` (**ProfileToken=profile1**) message to remove `VideoEncoderConfiguration` from profile1  
 The DUT sends `RemoveVideoEncoderConfigurationResponse` message
9. Execute steps 10-21  
 If there is no profile with "fixed" attribute value "false" in profiles list, end test.
10. If the DUT sends `CreateProfileResponse` message, validate `CreateProfileResponse` message from the DUT.
11. NVC invokes `AddVideoSourceConfigurationRequest` (**ConfigurationToken=first video source from the list on step 6 (VSC1), ProfileToken=NewProfile**) message to add `VideoSourceConfiguration` to the new profile.
12. The DUT sends `AddVideoSourceConfigurationResponse` message.
13. NVC invokes `AddVideoEncoderConfigurationRequest` (**ConfigurationToken=first video encoder from the list on step 4 (VEC1), ProfileToken=NewProfile**) message to add `VideoEncoderConfiguration` to the new profile.
14. The DUT sends `AddVideoEncoderConfigurationResponse` message.
15. NVC invokes `RemoveVideoEncoderConfigurationRequest` (**ProfileToken=NewProfile**) message to remove video encoder configuration from `NewProfileToken`.
16. The DUT sends `RemoveVideoEncoderConfigurationResponse` message.
17. NVC invokes `GetVideoEncoderConfigurationRequest` (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.
18. The DUT sends `GetVideoEncoderConfigurationResponse` message.
19. Check that **UseCount=usecount1**, in `GetVideoEncoderConfigurationResponse` where `usecount1` is `UseCount` value for `VEC1` in the list on step 4. If step 8 is executed, the `usecount1` could be reduced by 1.



20. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile.
21. The DUT sends DeleteProfileResponse message.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send RemoveVideoEncoderConfigurationResponse message.

The DUT did not send valid RemoveVideoEncoderConfigurationResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is not decreased by 1 after removing of the VideoEncoderConfiguration from profile.

#### 4.4.9 VIDEO ENCODER CONFIGURATION USE COUNT (PROFILE DELETION WITH VIDEO ENCODER CONFIGURATION)

**Test Label:** Media Service DUT Video Encoder Configuration Use Count Validation after Deletion of Profile with Video Encoder Configuration Validation.

**Test Case ID:** MEDIA-2-3-9

**ONVIF Core Specification Coverage:** Get video encoder configurations, Get video encoder configuration, Delete media profile

**Command Under Test:** GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, DeleteProfile

**WSDL Reference:** media.wsdl

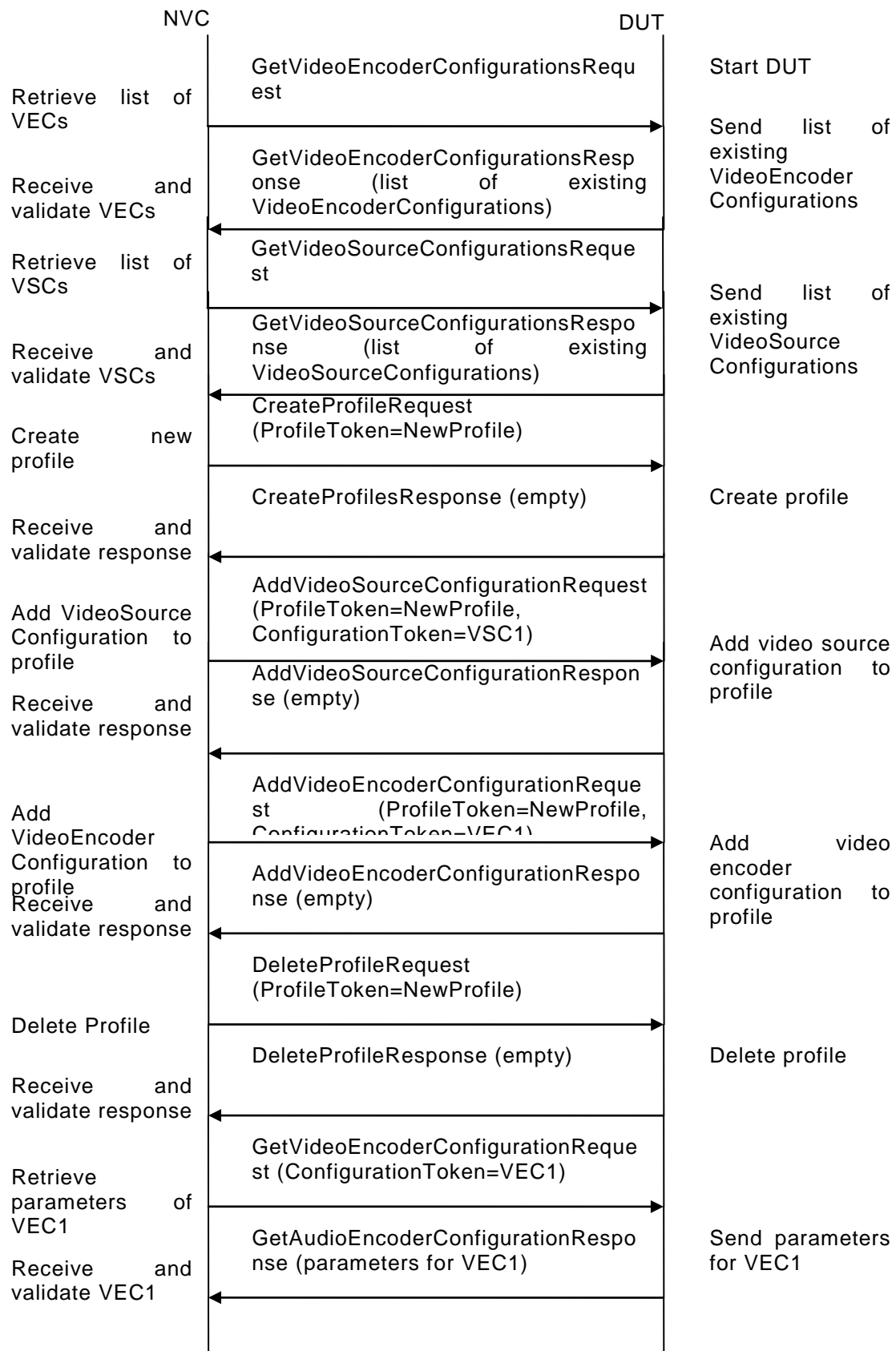
**Requirement Level:** MUST

**Test Propose:** To check Video Encoder Configuration use count after deletion of the profile with video encoder configuration

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` message to retrieve list of video encoder configurations from device.
4. The DUT sends `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `GetVideoSourceConfigurationsRequest` message to retrieve list of video source configurations from device.
6. The DUT sends `GetVideoSourceConfigurationsResponse` message.
7. NVC invokes `CreateProfileRequest` (**Name=NewName, Token=NewProfile**) message to create profile.
8. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
 If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
 NVC invokes `RemoveVideoEncoderConfigurationRequest` (**ProfileToken=profile1**) message to remove `VideoEncoderConfiguration` from profile1  
 The DUT sends `RemoveVideoEncoderConfigurationResponse` message  
 Execute steps 10-18  
 If there is no profile with "fixed" attribute value "false" in profiles list, end test.
9. If the DUT sends `CreateProfileResponse` message, validate `CreateProfileResponse` message from the DUT.
10. NVC invokes `AddVideoSourceConfigurationRequest` (**ConfigurationToken=first video source configuration from the list on step 6 (VSC1), ProfileToken=NewProfile**) message to add `VideoSourceConfiguration` to the new profile.
11. The DUT sends `AddVideoSourceConfigurationResponse` message.
12. NVC invokes `AddVideoEncoderConfigurationRequest` (**ConfigurationToken=first video encoder configuration from the list on step 4 (VEC1), ProfileToken=NewProfile**) message to add `VideoEncoderConfiguration` to the new profile.
13. The DUT sends `AddVideoEncoderConfigurationResponse` message.
14. NVC invokes `DeleteProfileRequest` (**ProfileToken=NewProfile**) message to remove profile with video encoder configuration.
15. The DUT sends `DeleteProfileResponse` message.
16. NVC invokes `GetVideoEncoderConfigurationRequest` (**ConfigurationToken=VEC1**) message to retrieve video encoder configuration parameters.
17. The DUT sends `GetVideoEncoderConfigurationResponse` message.
18. Check that **UseCount=usecount1**, in `GetVideoEncoderConfigurationResponse` where usecount1 is UseCount value for VEC1 in the list on step 4. If step 8 is executed, the usecount1 could be reduced by 1.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

The DUT did not send valid DeleteProfileResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the VideoEncoderConfiguration.

**4.4.10 VIDEO ENCODER CONFIGURATION USE COUNT (SET VIDEO ENCODER CONFIGURATION)**

**Test Label:** Media Service Video Encoder Configuration Use Count Validation after Changing of Video Encoder Configuration Validation.

**Test Case ID:** MEDIA-2-3-10

**ONVIF Core Specification Coverage:** Get video encoder configurations, Get video encoder configuration, Modify a video encoder configuration.

**Command Under Test:** GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, SetVideoEncoderConfiguration

**WSDL Reference:** media.wsdl

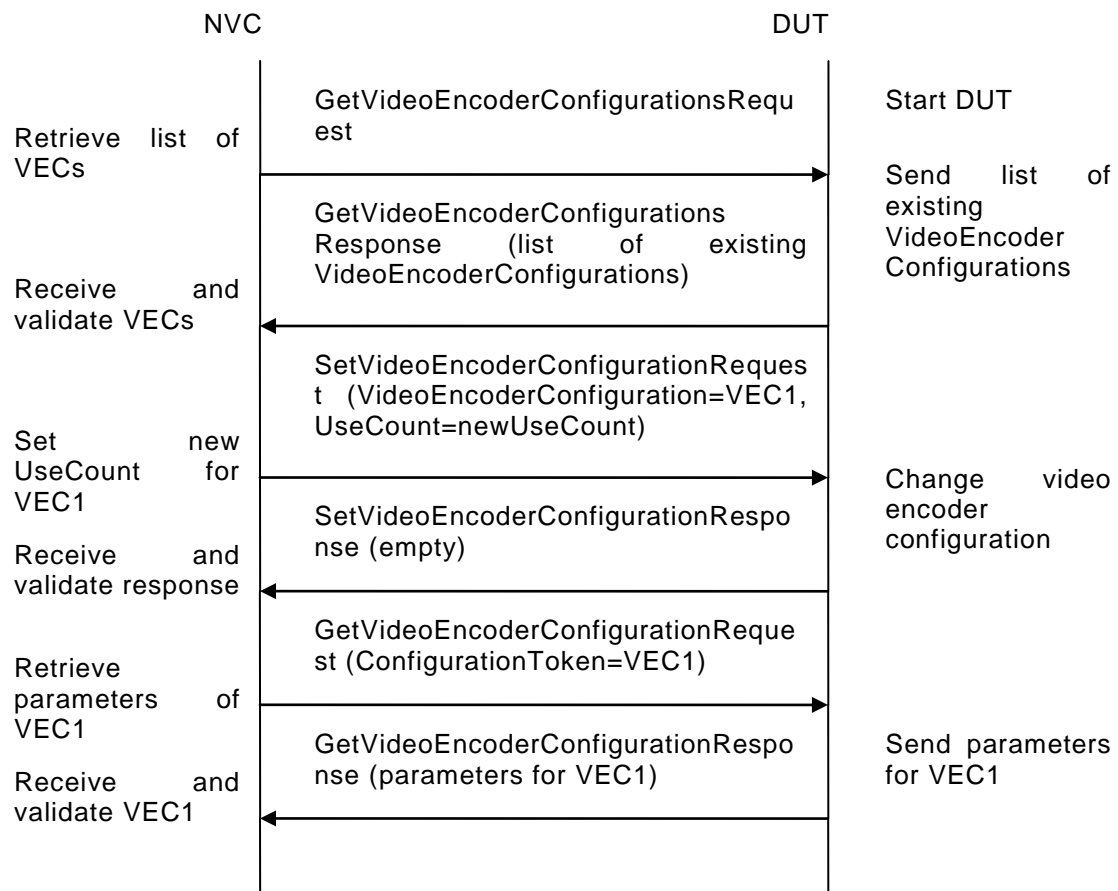
**Requirement Level:** MUST

**Test Propose:** To check Video Encoder Configuration use count after setting video encoder configuration parameters.

**Pre-Requisite:** Media is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetVideoEncoderConfigurationsRequest` message to retrieve list of video encoder configurations from device.
4. The DUT sends `GetVideoEncoderConfigurationsResponse` message.
5. NVC invokes `SetVideoEncoderConfigurationRequest` (token=the first Video Encoder Configuration from the list on step 4(VEC1), `UseCount=NewUseCount`) message to set parameters for video encoder configuration.
6. The DUT sends `SetVideoEncoderConfigurationResponse` message.
7. NVC invokes `GetVideoEncoderConfigurationRequest` (`ConfigurationToken=VEC1`) message to retrieve video encoder configuration parameters.
8. The DUT sends `GetVideoEncoderConfigurationResponse` message.
9. Check that `UseCount = usecount1`, in `GetVideoEncoderConfigurationResponse` where `usecount1` is `UseCount` value for VEC1 in the list on step 4.

### Test Result:

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send RemoveVideoEncoderConfigurationResponse message.

The DUT did not send valid RemoveVideoEncoderConfigurationResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is changed after trying to set UseCount value.

## **4.5 Audio Configuration**

### **4.5.1 AUDIO SOURCE CONFIGURATION**

**Test Label:** Media Configuration Audio Source Configuration

**Test Case ID:** MEDIA-3-1-1

**ONVIF Core Specification Coverage:** Create media profile, Add audio source configuration to a profile, Remove audio source configuration from a profile, Delete media profile, Get audio sources, Get audio source configurations, Get audio source configuration, Get compatible audio source configurations, Get audio source configuration options, Modify an audio source configuration.

**Command Under Test:** None

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF SUPPORTED (Audio)

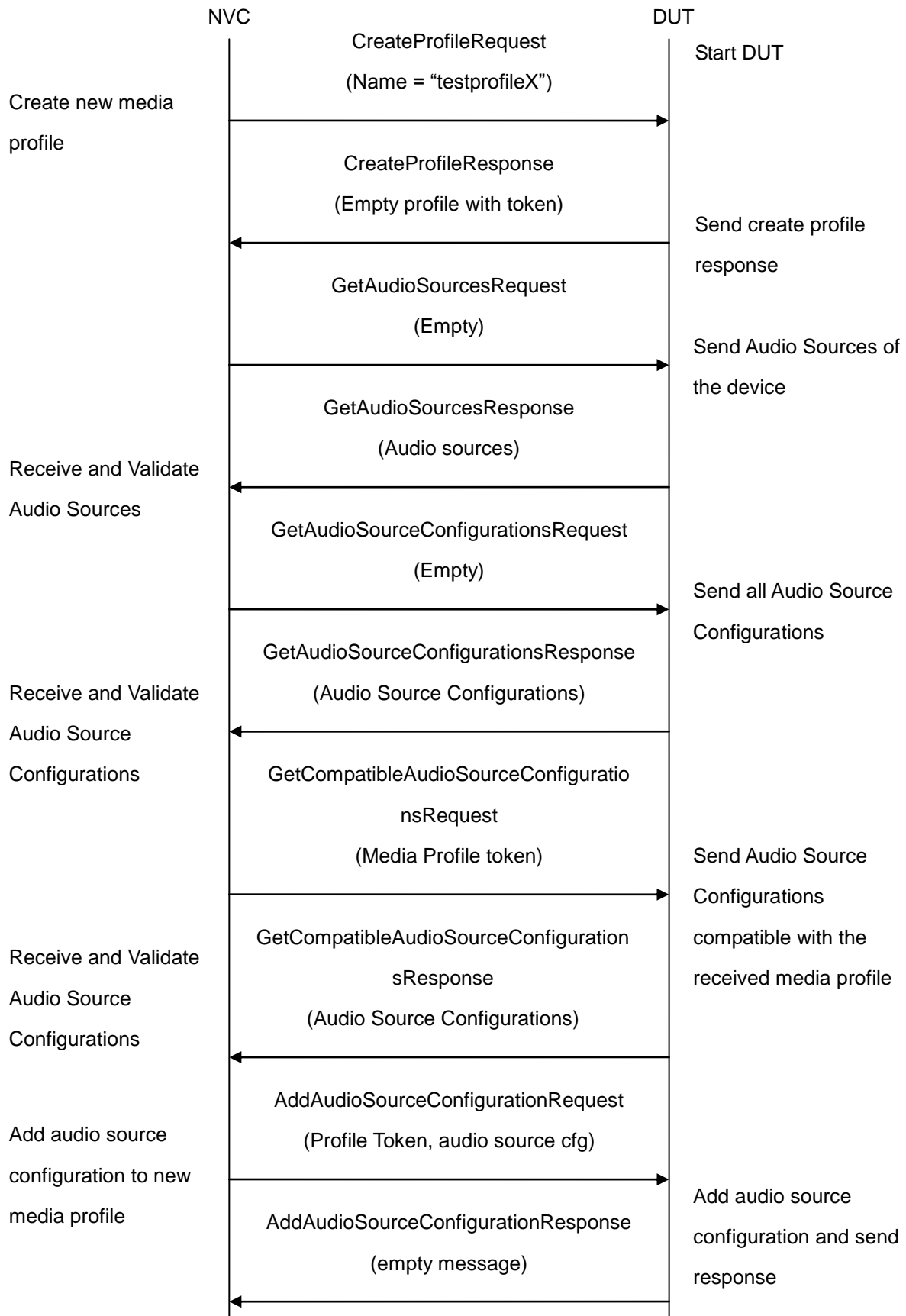
**Test Purpose:** To verify DUT Audio Source Configuration Operations

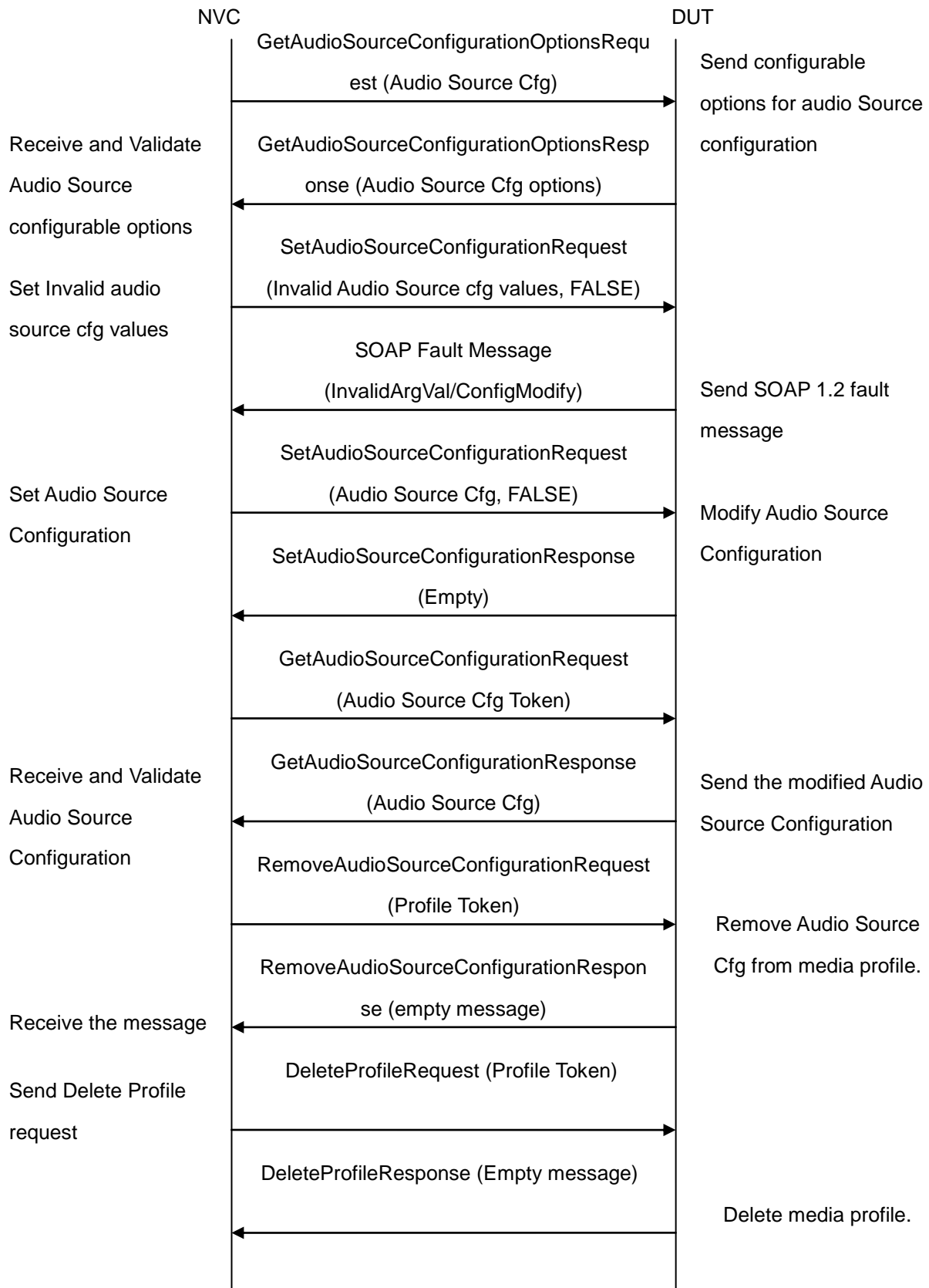
**Pre-Requisite:** Audio is supported by DUT

**Test Configuration:** NVC and DUT

**Test Sequence:**

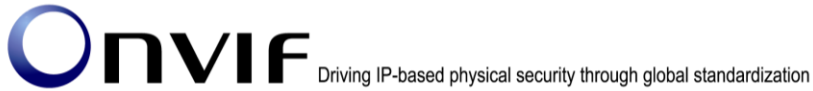






**Test Procedure:**

1. Start an NVC.
2. Start the DUT.
3. NVC invokes CreateProfile request with **ProfileToken** = '**testprofileX**'.
4. DUT creates new media profile and sends the response.
5. NVC invokes GetAudioSources request to retrieve the existing audio sources of DUT.
6. NVC verifies the list of audio sources sent by DUT.
7. NVC invokes GetAudioSourceConfigurations request to retrieve the list of audio source configurations supported by the DUT.
8. NVC verifies the list of audio source configurations sent by DUT.
9. NVC invokes GetCompatibleAudioSourceConfigurations request with '**testprofileX**' as **ProfileToken**.
10. DUT sends the list of audio source configurations compatible with the received media profile token.
11. NVC invokes AddAudioSourceConfiguration request message with **ProfileToken** as '**testprofileX**' and **ConfigurationToken** as one of the tokens in GetCompatibleAudioSourceConfigurations response.
12. DUT adds the audio source configuration to the media profile and sends the response.
13. NVC invokes GetAudioSourceConfigurationOptions request with **ConfigurationToken** as the same token sent in the AddAudioSourceConfiguration request.
14. DUT sends the configurable options supported for the received audio source configuration.
15. NVC invokes SetAudioSourceConfiguration request with audio source configuration values outside the range defined in GetAudioSourceConfigurationOptions response and '**ForcePersistence**' flag as '**FALSE**'.
16. DUT send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**).
17. NVC verifies the SOAP fault message sent by DUT.
18. NVC invokes SetAudioSourceConfiguration request with audio source configuration values as defined in GetAudioSourceConfigurationOptions response and '**ForcePersistence**' flag as '**FALSE**'.
19. DUT modifies the audio source configuration and sends the SetAudioSourceConfigurationResponse message indicating success.
20. NVC verifies the modified audio source configuration by invoking the GetAudioSourceConfiguration request.
21. DUT sends the modified audio source configuration in GetAudioSourceConfiguration response.
22. NVC invokes RemoveAudioSourceConfiguration request with **ProfileToken** as '**testprofileX**'.
23. DUT removes the audio source configuration token from media profile and sends the response.



24. NVC invokes DeleteProfile request with **ProfileToken** as '**testprofileX**'.

25. DUT deletes the media profile and sends the response.

**Test Result:**

**PASS -**

DUT passes all assertions.

**FAIL -**

DUT did not send CreateProfileResponse message.

DUT did not send valid GetAudioSourcesResponse message.

DUT did not send valid GetAudioSourceConfigurationsResponse message.

DUT did not send GetCompatibleAudioSourceConfigurationsResponse message.

DUT did not send AddAudioSourceConfigurationResponse message.

DUT did not send GetAudioSourceConfigurationOptionsResponse message.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioSourceConfiguration request.

DUT did not send SetAudioSourceConfigurationResponse message.

DUT did not send GetAudioSourceConfigurationResponse message.

DUT did not modify audio source configuration correctly.

DUT did not send RemoveAudioSourceConfigurationResponse message.

DUT did not send DeleteProfileResponse message.

#### **4.5.2 AUDIO ENCODER CONFIGURATION**

**Test Label:** Media Configuration Audio Encoder Configuration

**Test Case ID:** MEDIA-3-1-2

**ONVIF Core Specification Coverage:** Create media profile, Add audio source configuration to a profile, Add audio encoder configuration to a profile, Remove audio source configuration from a profile, Remove audio encoder configuration from a profile, Delete media profile, Get audio source configurations, Get audio encoder configurations, Get compatible audio encoder configurations.

**Command Under Test:** None

**WSDL Reference:** media.wsdl

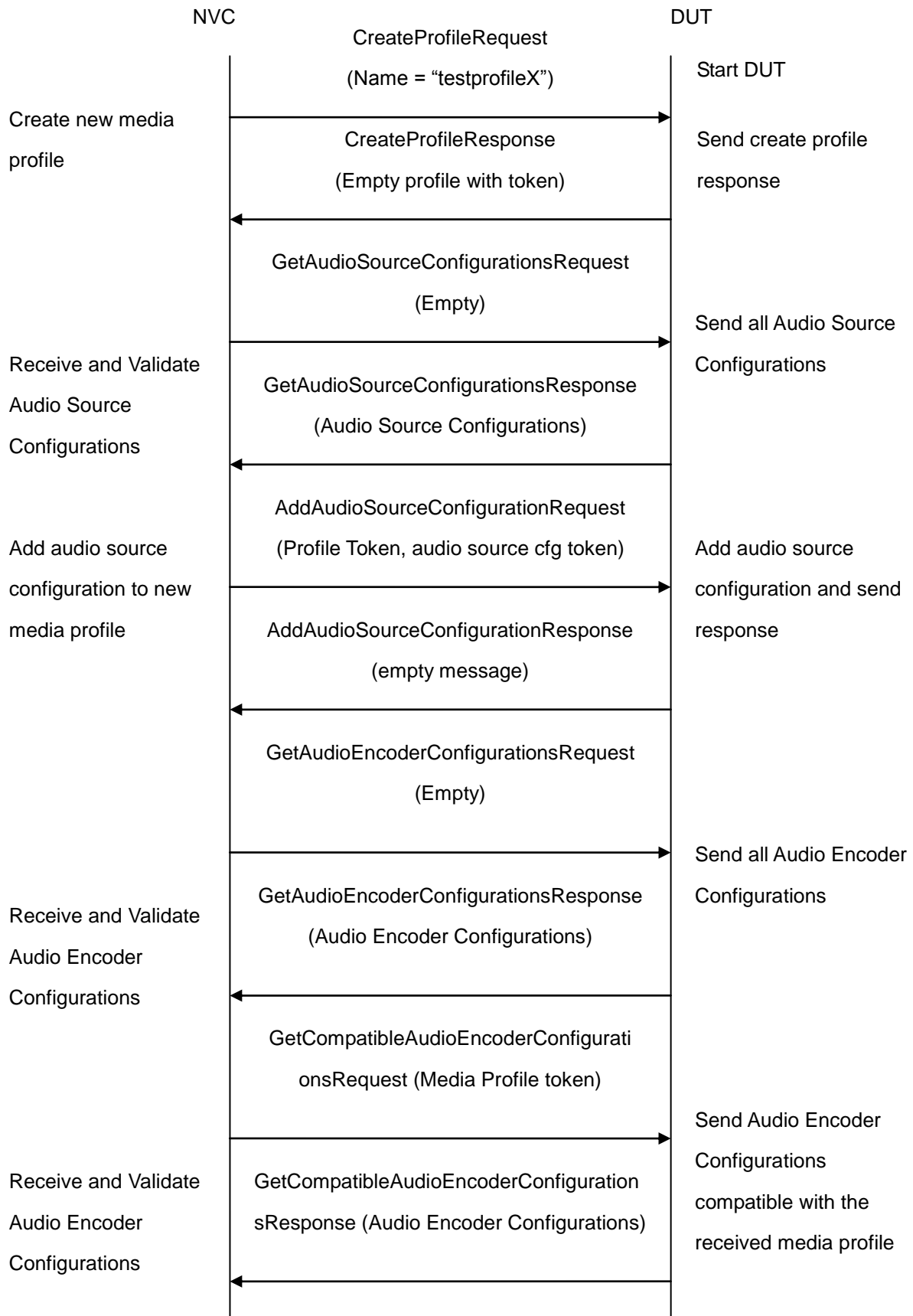
**Requirement Level:** MUST IF SUPPORTED (Audio)

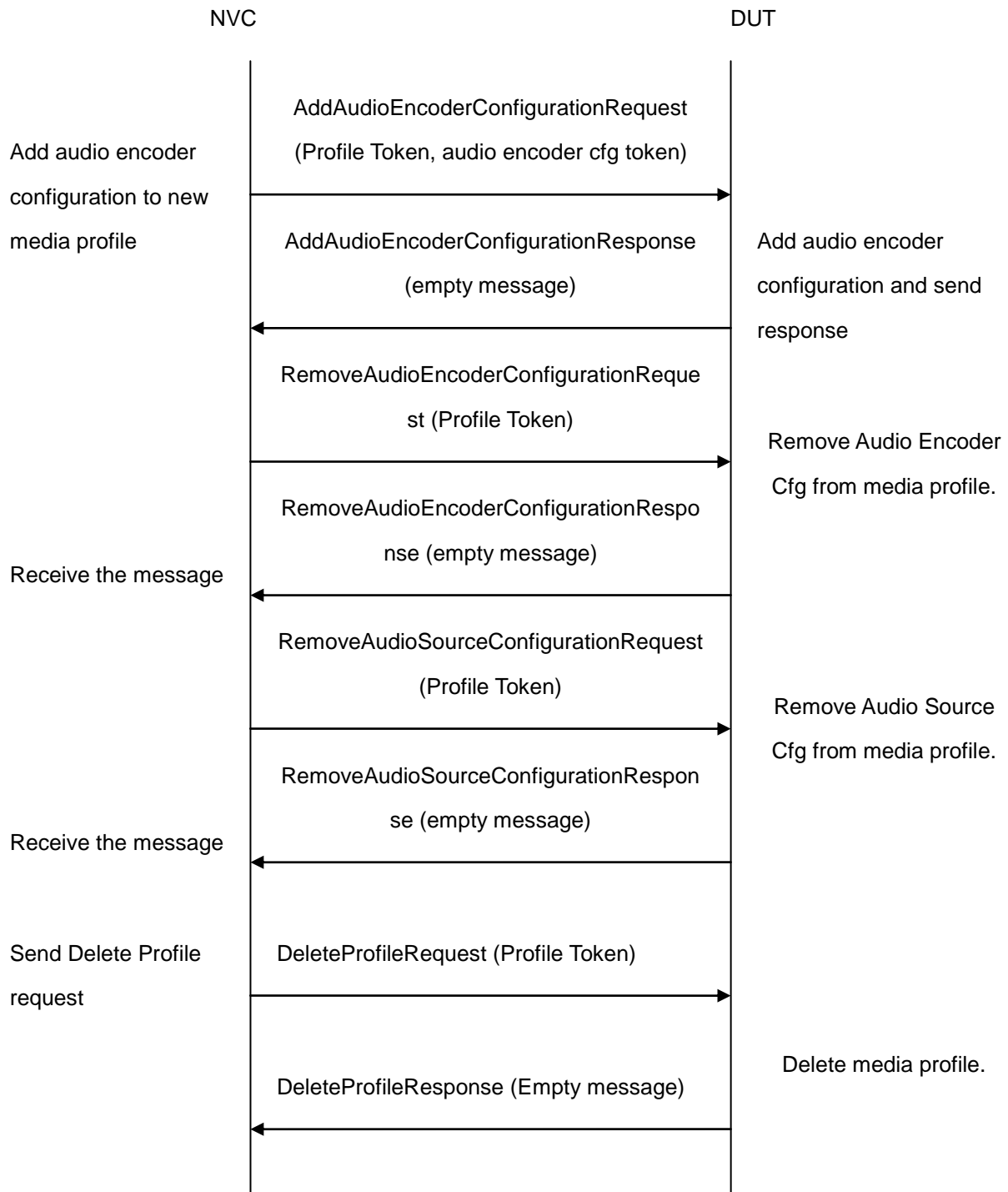
**Test Purpose:** To verify DUT Audio Encoder Configuration Operations

**Pre-Requisite:** Audio is supported by DUT

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes CreateProfile request with **ProfileToken** = 'testprofileX'.

4. DUT creates new media profile and sends the response.
5. NVC will invoke GetAudioSourceConfigurations request to retrieve the list of audio source configurations supported by DUT.
6. NVC verifies the list of audio source configurations sent by DUT.
7. NVC invokes AddAudioSourceConfiguration request message with **ProfileToken** as 'testprofileX' and **ConfigurationToken** as one of the tokens received in the GetAudioSourceConfigurations response.
8. DUT adds the audio source configuration to the profile and sends the response.
9. NVC will invoke GetAudioEncoderConfigurations request to retrieve the list of audio encoder configurations supported by DUT.
10. NVC verifies the list of audio encoder configurations sent by DUT.
11. NVC invokes GetCompatibleAudioEncoderConfigurations request with 'testprofileX' as **ProfileToken**.
12. DUT sends the list of audio encoder configurations compatible with the received media profile token.
13. NVC invokes AddAudioEncoderConfiguration request message with **ProfileToken** as 'testprofileX' and **ConfigurationToken** as one of the tokens received in the GetCompatibleAudioEncoderConfigurations response.
14. DUT adds the audio encoder configuration to the profile and sends the response.
15. NVC invokes RemoveAudioEncoderConfiguration request with **ProfileToken** as 'testprofileX'.
16. DUT removes the audio encoder configuration token from media profile and sends the response.
17. NVC invokes RemoveAudioSourceConfiguration request with **ProfileToken** as 'testprofileX'.
18. DUT removes the audio source configuration token from media profile and sends the response.
19. NVC invokes DeleteProfile request with **ProfileToken** as 'testprofileX'.
20. DUT deletes the media profile and sends the response.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send CreateProfileResponse message.

DUT did not send valid GetAudioSourceConfigurationsResponse message.

DUT did not send AddAudioSourceConfigurationResponse message.

DUT did not send valid GetAudioEncoderConfigurationsResponse message.



DUT did not send GetCompatibleAudioEncoderConfigurationsResponse message.

DUT did not send AddAudioEncoderConfigurationResponse message.

DUT did not send RemoveAudioEncoderConfigurationResponse message.

DUT did not send RemoveAudioSourceConfigurationResponse message.

DUT did not send DeleteProfileResponse message.

#### **4.5.3 G.711 AUDIO ENCODER CONFIGURATION**

**Test Label:** Media Configuration DUT G.711 Audio Encoder Configuration

**Test Case ID:** MEDIA-3-1-3

**ONVIF Core Specification Coverage:** Get audio encoder configurations, Get audio encoder configuration, Get audio encoder configuration options, Modify audio encoder configurations

**Command Under Test:** None

**WSDL Reference:** media.wsdl

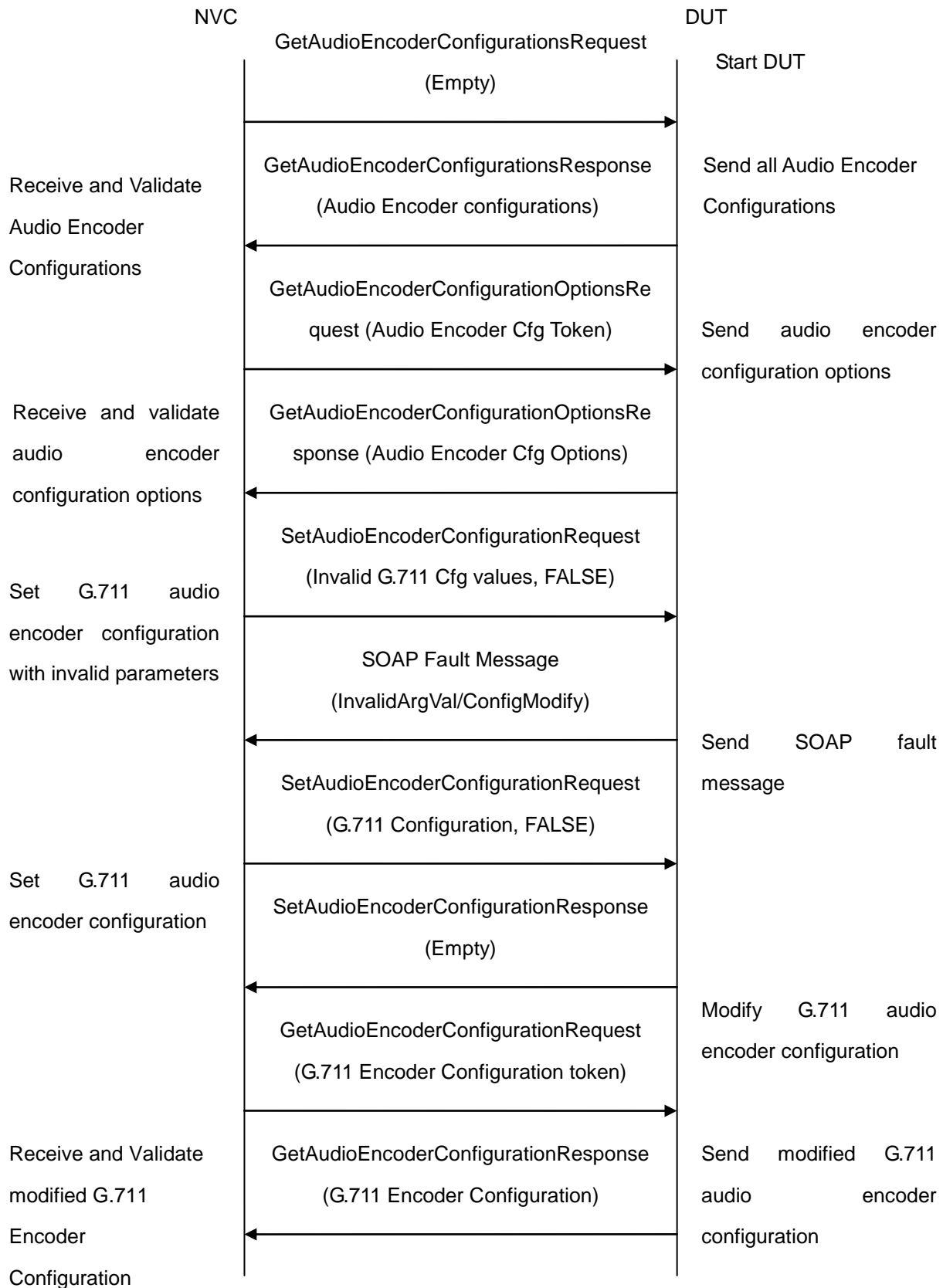
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Purpose:** To verify DUT G.711 Audio Encoder Configuration Setting

**Pre-Requisite:** Audio is supported by DUT

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes GetAudioEncoderConfigurations request.
4. DUT sends the list of supported audio encoder configurations in GetAudioEncoderConfigurationsResponse message.
5. NVC invokes GetAudioEncoderConfigurationOptions Request (Audio Encoder Configuration token) to retrieve audio encoder configuration options for the specified audio encoder configuration.
6. DUT sends the range of configurable values for the received audio encoder configuration in the GetAudioEncoderConfigurationOptionsResponse message.
7. Test steps -5 & 6 have to be repeated for all audio encoder configurations until NVC finds a audio encoder configuration with G.711 encoding support
8. NVC invokes SetAudioEncoderConfiguration request with G.711 configuration values outside the range defined in the GetAudioEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
9. DUT send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**).
10. NVC verifies the SOAP fault message sent by DUT.
11. NVC invokes SetAudioEncoderConfiguration request (**Encoding = "G711", Bit Rate = r1, Sample Rate = r2 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptionsResponse message.
12. DUT modifies G.711 audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
13. NVC verifies the G.711 audio Encoder Configuration settings on DUT by invoking GetAudioEncoderConfiguration request.
14. DUT sends modified G.711 audio Encoder Configuration in the GetAudioEncoderConfigurationResponse message (**Encoding = "G711", Bit Rate = r1, Sample Rate = r2**).
- 15.

### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not send GetAudioEncoderConfigurationsResponse message.

DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

DUT doesn't support G.711 audio encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioEncoderConfiguration request.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not modify G.711 Audio Encoder Configuration.

#### **4.5.4 G.726 AUDIO ENCODER CONFIGURATION**

**Test Label:** Media Configuration DUT G.726 Audio Encoder Configuration

**Test Case ID:** MEDIA-3-1-4

**ONVIF Core Specification Coverage:** Get audio encoder configurations, Get audio encoder configuration, Get audio encoder configuration options, Modify audio encoder configurations

**Command Under Test:** None

**WSDL Reference:** media.wsdl

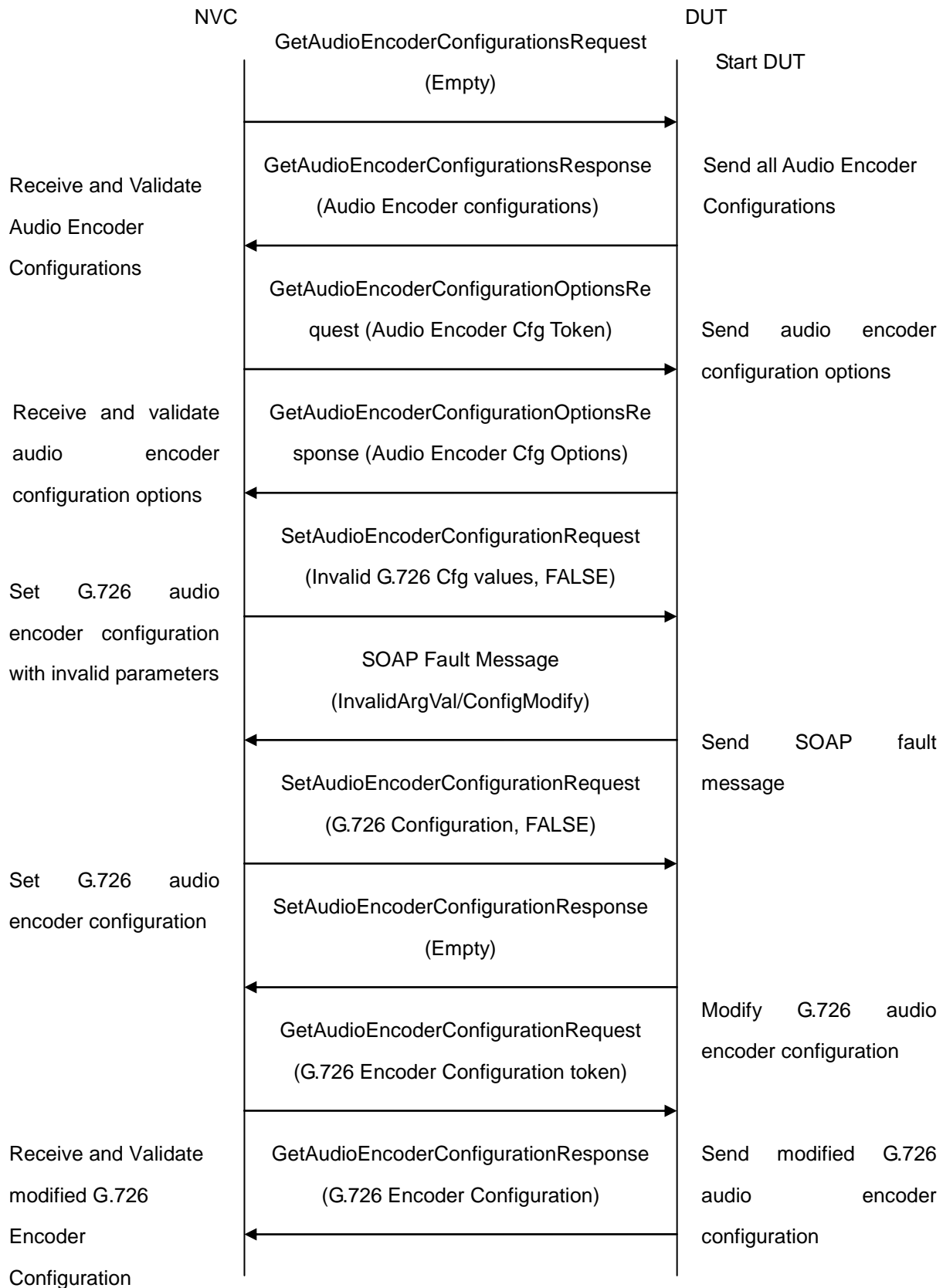
**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (G.726)

**Test Purpose:** To verify DUT G.726 Audio Encoder Configuration Setting

**Pre-Requisite:** Audio is supported by DUT and G.726 is implemented by DUT.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes GetAudioEncoderConfigurations request.
4. DUT sends the list of supported audio encoder configurations in GetAudioEncoderConfigurationsResponse message.
5. NVC invokes GetAudioEncoderConfigurationOptions Request (Audio Encoder Configuration token) to retrieve audio encoder configuration options for the specified audio encoder configuration.
6. DUT sends the range of configurable values for the received audio encoder configuration in the GetAudioEncoderConfigurationOptionsResponse message.
7. Test steps -5 & 6 have to be repeated for all audio encoder configurations until NVC finds a audio encoder configuration with G.726 encoding support
8. NVC invokes SetAudioEncoderConfiguration request with G.726 configuration values outside the range defined in the GetAudioEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
9. DUT send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**).
10. NVC verifies the SOAP fault message sent by DUT.
11. NVC invokes SetAudioEncoderConfiguration request (**Encoding = "G726", Bit Rate = r1, Sample Rate = r2 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptionsResponse message.
12. DUT modifies G.726 audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
13. NVC verifies the G.726 audio Encoder Configuration settings on DUT by invoking GetAudioEncoderConfiguration request.
14. DUT sends the modified G.726 audio Encoder Configuration in the GetAudioEncoderConfigurationResponse message (**Encoding = "G726", Bit Rate = r1, Sample Rate = r2**).

### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not send GetAudioEncoderConfigurationsResponse message.

DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

DUT doesn't support G.726 audio encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioEncoderConfiguration request.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not modify G.726 Audio Encoder Configuration.

#### **4.5.5 AAC AUDIO ENCODER CONFIGURATION**

**Test Label:** Media Configuration AAC Audio Encoder Configuration

**Test Case ID:** MEDIA-3-1-5

**ONVIF Core Specification Coverage:** Get audio encoder configurations, Get audio encoder configuration, Get audio encoder configuration options, Modify audio encoder configurations

**Command Under Test:** None

**WSDL Reference:** media.wsdl

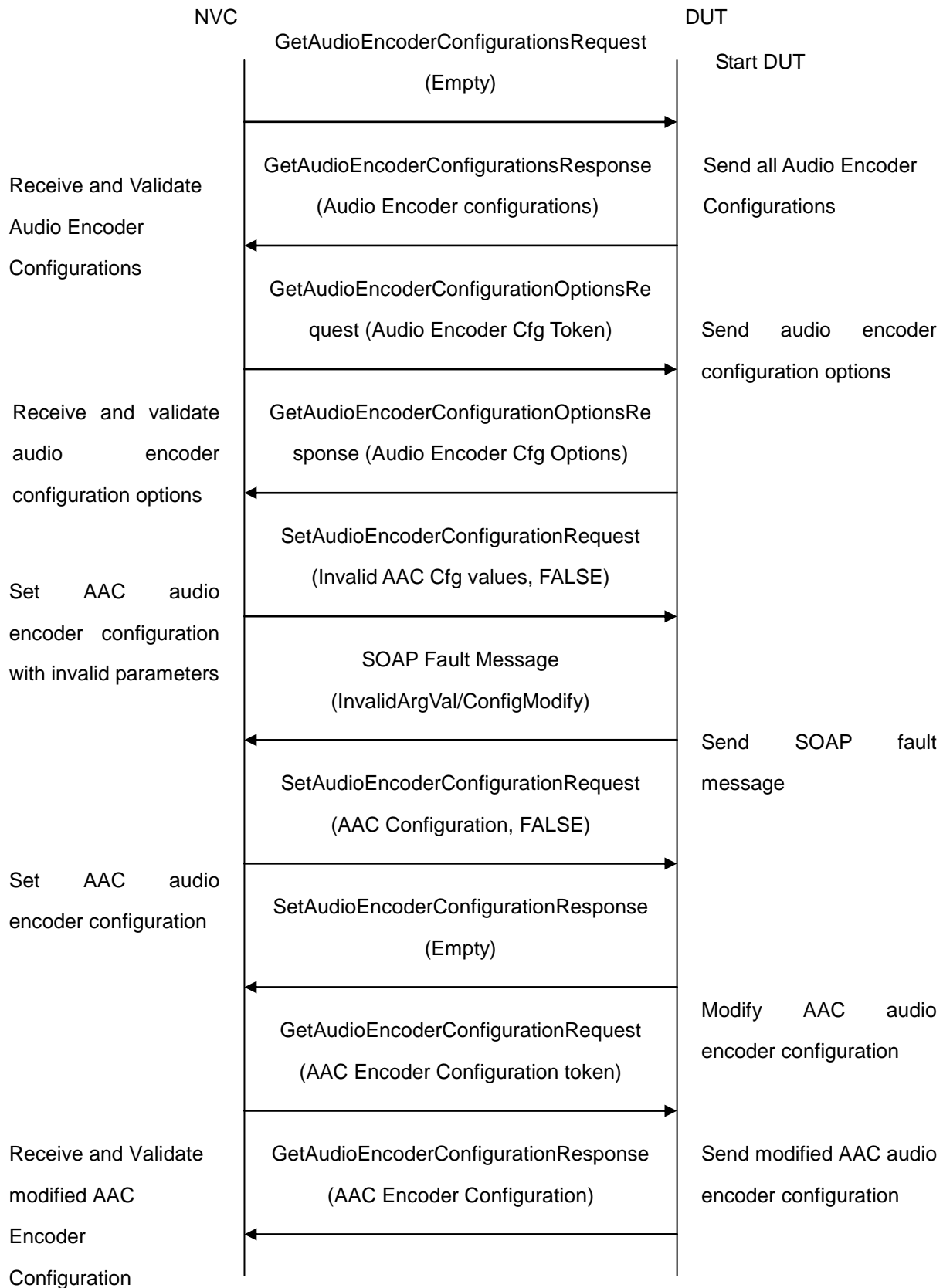
**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (AAC)

**Test Purpose:** To verify DUT AAC Audio Encoder Configuration Setting

**Pre-Requisite:** Audio is supported by DUT and AAC is implemented by DUT.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC invokes GetAudioEncoderConfigurations request.
4. DUT sends the list of supported audio encoder configurations in GetAudioEncoderConfigurationsResponse message.
5. NVC invokes GetAudioEncoderConfigurationOptions Request (Audio Encoder Configuration token) to retrieve audio encoder configuration options for the specified audio encoder configuration.
6. DUT sends the range of configurable values for the received audio encoder configuration in the GetAudioEncoderConfigurationOptionsResponse message.
7. Test steps -5 & 6 have to be repeated for all audio encoder configurations until NVC finds a audio encoder configuration with AAC encoding support
8. NVC invokes SetAudioEncoderConfiguration request with AAC configuration values outside the range defined in the GetAudioEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
9. DUT send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**).
10. NVC verifies the SOAP fault message sent by DUT.
11. NVC invokes SetAudioEncoderConfiguration request (**Encoding = "AAC", Bit Rate = r1, Sample Rate = r2 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptionsResponse message.
12. DUT modifies AAC audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
13. NVC verifies the AAC audio Encoder Configuration settings on DUT by invoking GetAudioEncoderConfiguration request.
14. DUT sends modified AAC audio Encoder Configuration in the GetAudioEncoderConfigurationResponse message (**Encoding = "AAC", Bit Rate = r1, Sample Rate = r2**).

### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not send GetAudioEncoderConfigurationsResponse message.

DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

DUT doesn't support AAC audio encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioEncoderConfiguration request.



DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not modify AAC Audio Encoder Configuration.

#### 4.5.6 GET AUDIO SOURCE CONFIGURATION – INVALID CONFIGURATIONTOKEN

**Test Label:** Media Service DUT GetAudioSourceConfiguration Command Validation (Invalid Configuration Token).

**Test Case ID:** MEDIA-3-1-6

**ONVIF Core Specification Coverage:** Get audio source configuration

**Command Under Test:** GetAudioSourceConfiguration

**WSDL Reference:** media.wsdl

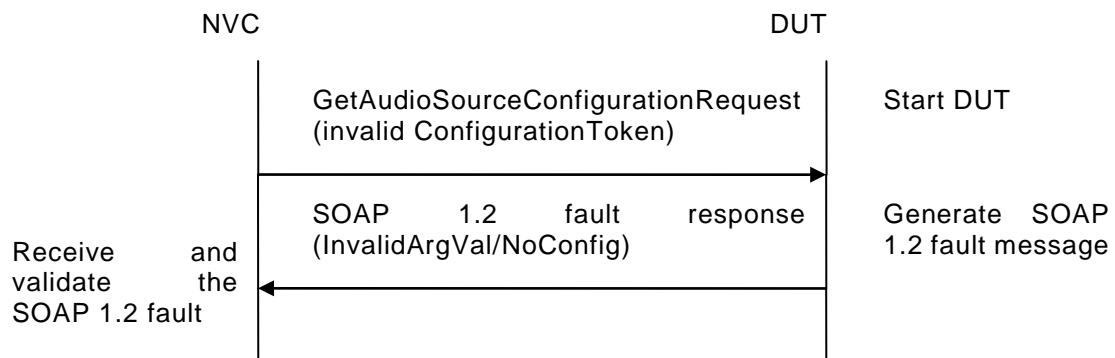
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To verify the behavior of GetAudioSourceConfiguration command in case of invalid ConfigurationToken.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationRequest (**invalid ConfigurationToken**) message.
4. The DUT will generate a SOAP 1.2 fault message (**InvalidArgVal/NoConfig**).

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

**Note:** Other faults than specified in the test are acceptable but specified is preferable.

**4.5.7 GET AUDIO SOURCE CONFIGURATION OPTIONS**

**Test Label:** Media Service DUT GetAudioSourceConfigurationOptions Command Validation.

**Test Case ID:** MEDIA-3-1-7

**ONVIF Core Specification Coverage:** Get audio source configuration options

**Command Under Test:** GetAudioSourceConfigurationOptions

**WSDL Reference:** media.wsdl

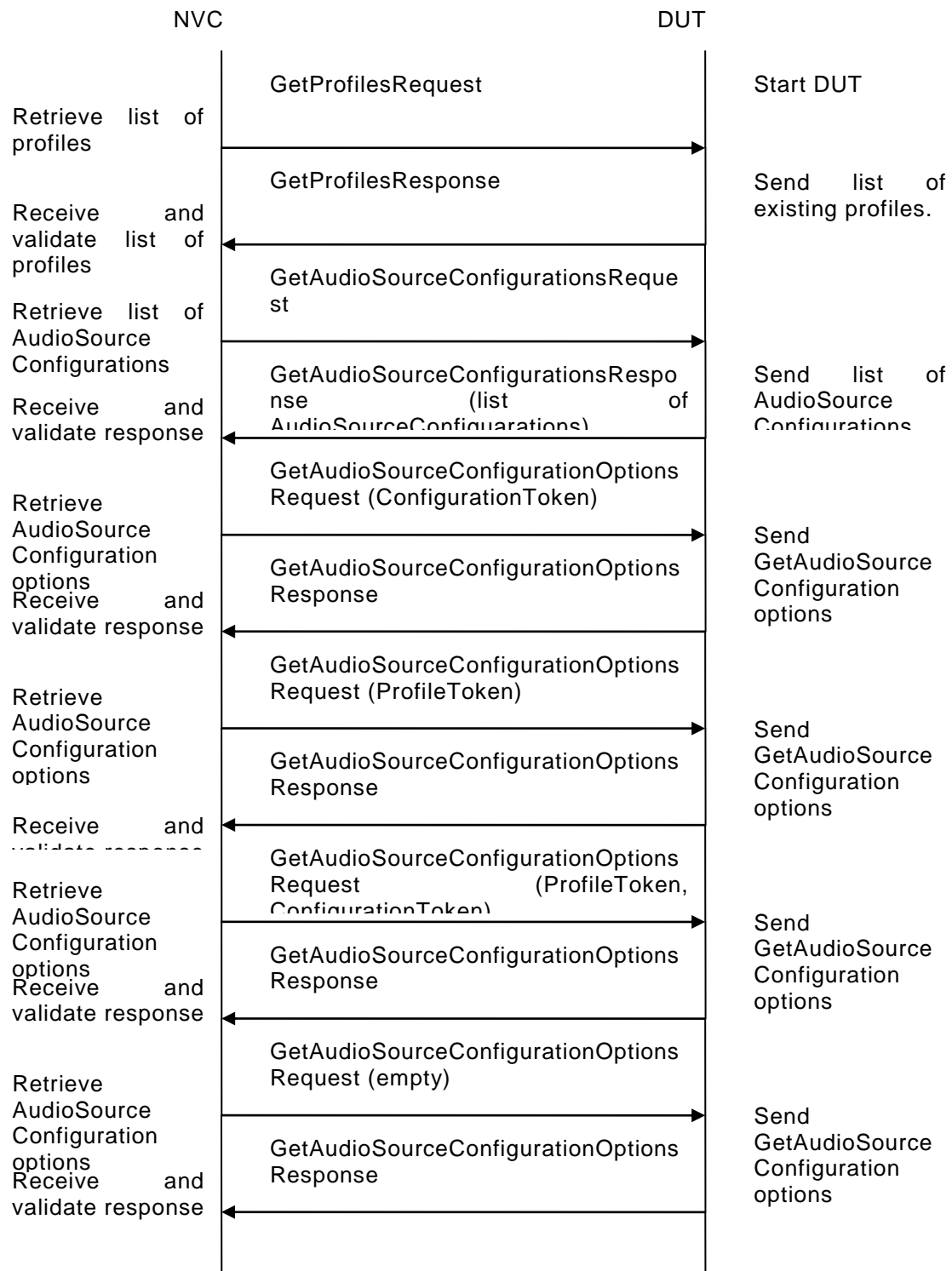
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To verify the behavior of GetAudioSourceConfigurationOptions command with different parameters.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.

2. Start a DUT.
3. NVC invokes GetProfilesRequest message to get list of existing profiles from device.
4. The DUT sends GetProfilesResponse message.
5. NVC invokes GetAudioSourceConfigurationsRequest message to get list of existing audio source configurations.
6. The DUT sends GetAudioSourceConfigurationsResponse message.
7. NVC invokes GetAudioSourceConfigurationOptionsRequest (**ConfigurationToken**) message.
8. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
9. Verify GetAudioSourceConfigurationOptionsResponse message.
10. NVC invokes GetAudioSourceConfigurationOptionsRequest (**ProfileToken**) message.
11. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
12. Verify GetAudioSourceConfigurationOptionsResponse message.
13. NVC invokes GetAudioSourceConfigurationOptionsRequest (**ProfileToken, ConfigurationToken**) message.
14. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
15. Verify GetAudioSourceConfigurationOptionsResponse message.
16. NVC invokes GetAudioSourceConfigurationOptionsRequest (empty) message.
17. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
18. Verify GetAudioSourceConfigurationOptionsResponse message.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.



The DUT did not send GetAudioSourcesConfigurationsResponse message.

The DUT did not send valid GetAudioSourcesConfigurationsResponse message.

The DUT did not send GetAudioSourcesConfigurationOptionsResponse message.

The DUT did not send valid GetAudioSourceConfigurationOptionsResponse message for one or more GetAudioSourceConfigurationOptionsRequest messages.

#### 4.5.8 GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID PROFILETOKEN

**Test Label:** Media Service DUT GetAudioSourceConfigurationOptions Command Validation (Invalid Profile Token).

**Test Case ID:** MEDIA-3-1-8

**ONVIF Core Specification Coverage:** Get audio source configuration options

**Command Under Test:** GetAudioSourceConfigurationOptions

**WSDL Reference:** media.wsdl

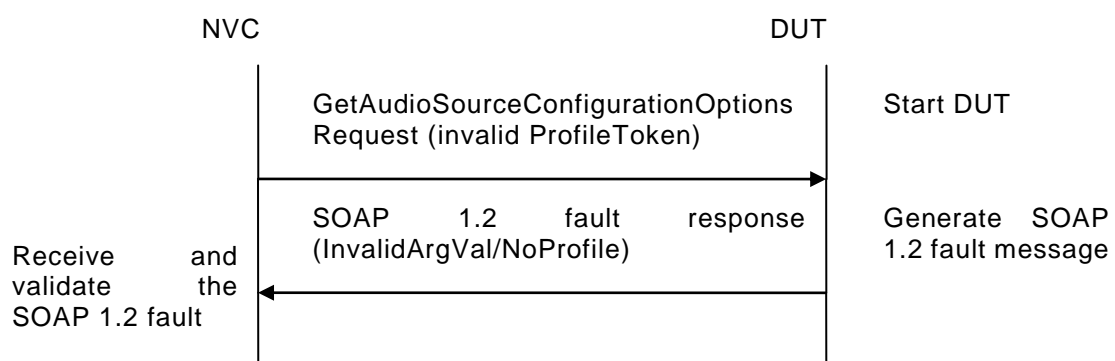
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To verify the behavior of GetAudioSourceConfigurationOptions command in case of invalid ProfileToken.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationOptionsRequest (**invalid ProfileToken**) message.

4. The DUT will generate a SOAP 1.2 fault message (**InvalidArgVal/NoProfile**).

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

**Note:** Other faults than specified in the test are acceptable but specified is preferable.

#### 4.5.9 GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID CONFIGURATION TOKEN

**Test Label:** Media Service DUT GetAudioSourceConfigurationOptions Command Validation (Invalid ConfigurationToken).

**Test Case ID:** MEDIA-3-1-9

**ONVIF Core Specification Coverage:** Get audio source configuration options

**Command Under Test:** GetAudioSourceConfigurationOptions

**WSDL Reference:** media.wsdl

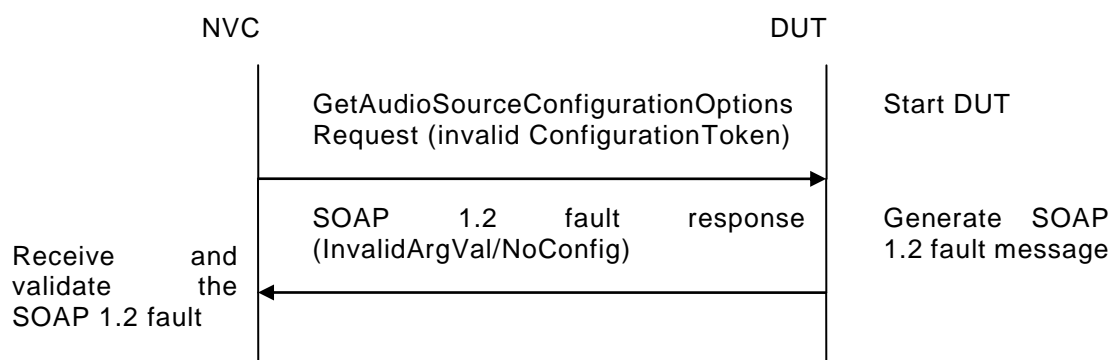
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To verify the behavior of GetAudioSourceConfigurationOptions command in case of invalid ConfigurationToken.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioSourceConfigurationOptionsRequest` (**invalid ConfigurationToken**) message.
4. The DUT will generate a SOAP 1.2 fault message (**InvalidArgVal/NoConfig**).

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

**Note:** Other faults than specified in the test are acceptable but specified is preferable.

**4.5.10 SET AUDIO SOURCE CONFIGURATION – INVALID TOKEN**

**Test Label:** Media Service DUT `SetAudioSourceConfiguration` Command Validation (Invalid Token).

**Test Case ID:** MEDIA-3-1-10

**ONVIF Core Specification Coverage:** Modify an audio source configuration

**Command Under Test:** `SetAudioSourceConfiguration`

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF SUPPORTED (Audio)

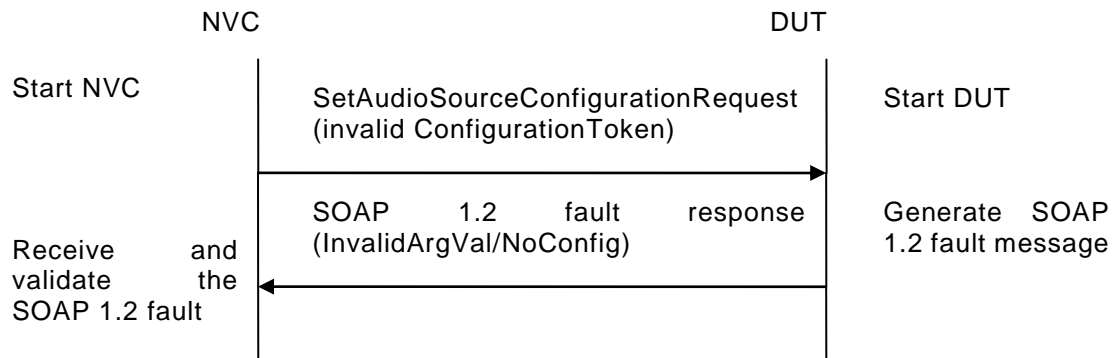
**Test Propose:** To verify the behavior of `SetAudioSourceConfiguration` command in case of invalid `ConfigurationToken`.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by `GetCapabilities` command.

**Test Configuration:** NVC and DUT

**Test Sequence:**




**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes SetAudioSourceConfigurationRequest (**invalid ConfigurationToken**) message.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoConfig).

**Test Result:**
**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

**Note:** Other faults than specified in the test are acceptable but specified is preferable.

## 4.6 Audio Source Configuration

### 4.6.1 AUDIO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY

**Test Label:** Media Service DUT GetAudioSourceConfigurations Command and GetProfiles Command Consistency Validation.

**Test Case ID:** MEDIA-3-2-1

**ONVIF Core Specification Coverage:** Get media profiles, Get audio source configurations

**Command Under Test:** GetProfiles, GetAudioSourceConfigurations

**WSDL Reference:** media.wsdl

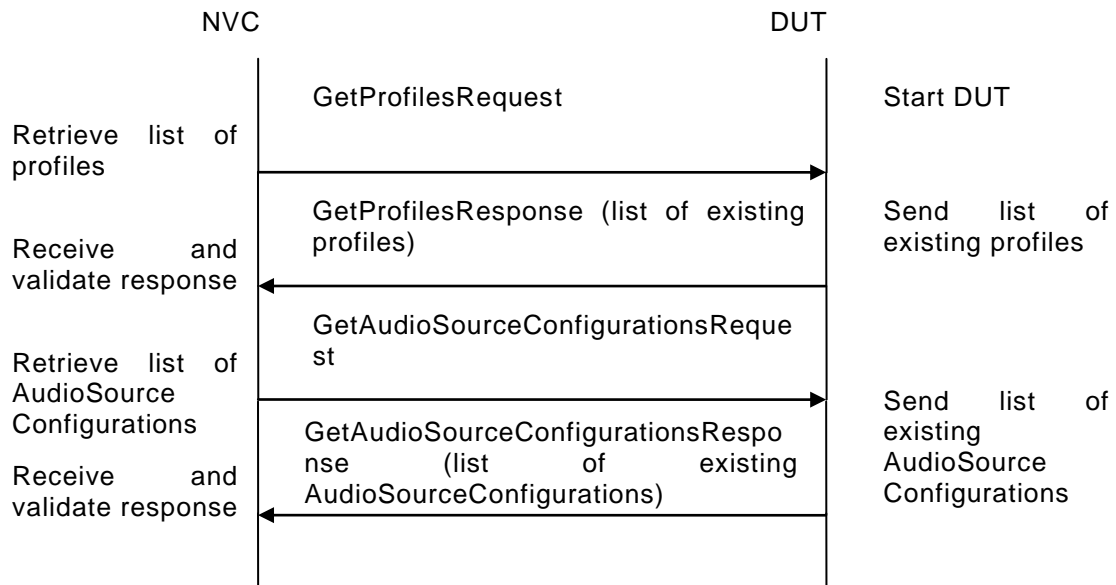
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetAudioSourceConfigurations command and GetProfiles command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetProfilesRequest message to retrieve list of profiles and their configurations.
4. The DUT sends GetProfilesResponse message.
5. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
6. The DUT sends GetAudioSourceConfigurationsResponse message.
7. Check that each AudioSourceConfiguration from GetAudioSourceConfigurationsResponse message has unique token.
8. Check that each AudioSourceConfiguration from the GetProfilesResponse message are included in the GetAudioSourceConfigurationsResponse message.
9. Check that AudioSourceConfiguration parameters are same in the GetProfilesResponse message and in the GetAudioSourceConfigurationsResponse message for each AudioSourceConfiguration.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT return two or more AudioSourceConfiguration in GetAudioSourceConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with AudioSourceConfigurations that were not included in the GetAudioSourceConfigurationsResponse message.

The DUT returned different parameters list and parameters values in the GetAudioSourceConfigurationsResponse message and in the GetProfilesResponse message for the same AudioSourceConfiguration.

#### **4.6.2 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION CONSISTENCY**

**Test Label:** Media Service DUT GetAudioSourceConfigurations Command and GetAudioSourceConfiguration Command Consistency Validation.

**Test Case ID:** MEDIA-3-2-2

**ONVIF Core Specification Coverage:** Get audio source configurations, Get audio source configuration

**Command Under Test:** GetAudioSourceConfigurations, GetAudioSourceConfiguration

**WSDL Reference:** media.wsdl

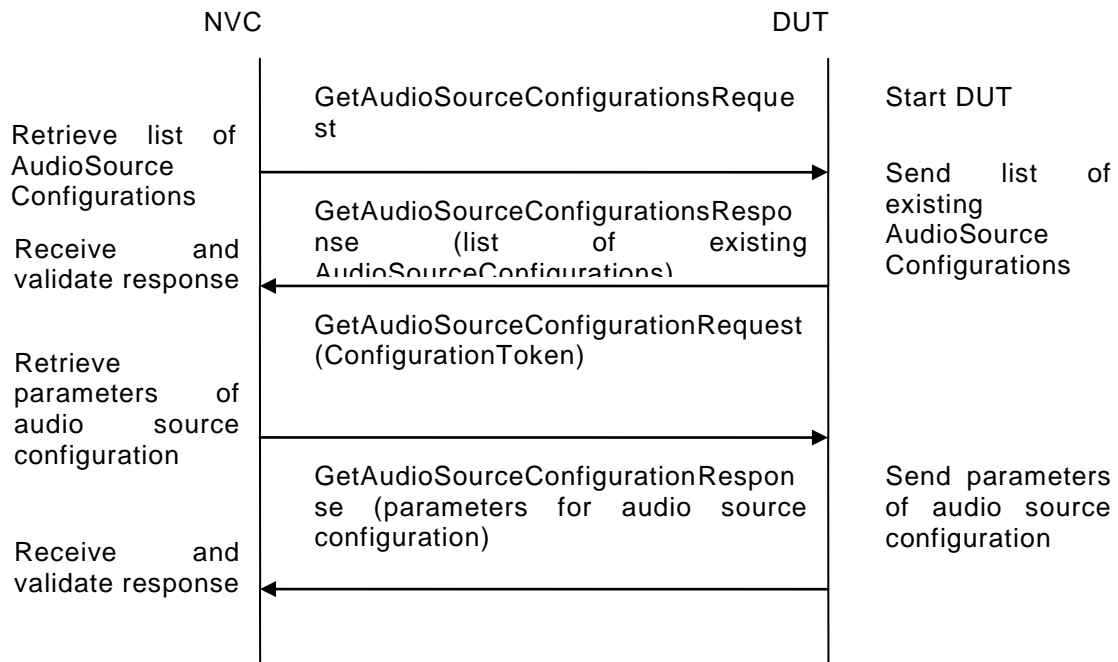
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetAudioSourceConfigurations command and GetAudioSourceConfiguration command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioSourceConfigurationsRequest` message to retrieve list of Audio Source Configurations from device.
4. The DUT sends `GetAudioSourceConfigurationsResponse` message.
5. NVC invokes `GetAudioSourceConfigurationRequest` (**ConfigurationToken**) message to retrieve parameters of Audio Source Configuration from device.
6. The DUT sends `GetAudioSourceConfigurationResponse` message.
7. Verify that all parameters and their values for `AudioSourceConfigurationToken` from `GetAudioSourceConfigurationsResponse` message and `GetAudioSourceConfigurationResponse` message are same.
8. Repeat steps 5-7 for other `AudioSourceConfigurations` from the `GetAudioSourceConfigurationsResponse` message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send `GetAudioSourceConfigurationsResponse` message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

The DUT did not send equal parameters for AudioSourceConfiguration in the GetAudioSourceConfigurationResponse message and in the GetAudioSourceConfigurationsResponse message.

#### 4.6.3 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION OPTIONS

##### CONSISTENCY

**Test Label:** Media Service DUT GetAudioSourceConfigurations Command and GetAudioSourceConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-3-2-3

**ONVIF Core Specification Coverage:** Get audio source configurations, Get audio source configuration options

**Command Under Test:** GetAudioSourceConfigurations, GetAudioSourceConfigurationOptions

**WSDL Reference:** media.wsdl

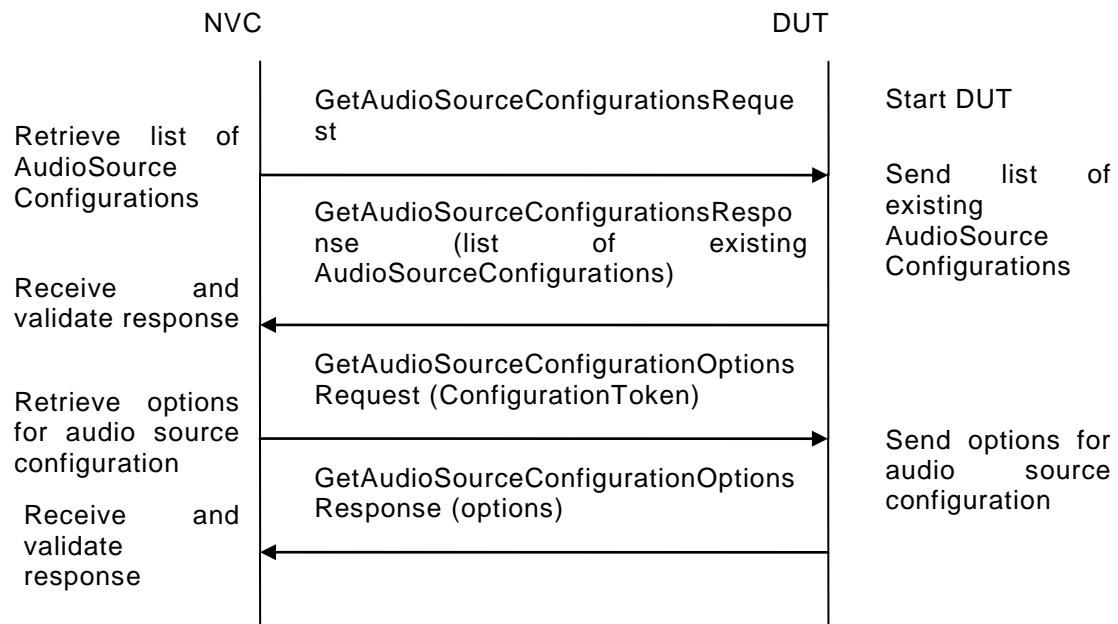
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetAudioSourceConfigurations command and AudioSourceConfigurationOptions command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. NVC invokes GetAudioSourceConfigurationOptionsRequest (**ConfigurationToken**) message to retrieve available audio source configuration options from device.
6. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
7. Check that InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message contains only unique tokens.
8. Check that SourceToken for the audio source configuration from GetAudioSourceConfigurationsResponse message exists in InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message.
9. Repeat steps 5-8 for each audio source configuration from the GetAudioSourceConfigurationsResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationOptionsResponse message.

The DUT did not send valid GetAudioSourceConfigurationOptionsResponse message.

The DUT sent InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message that contains not unique tokens.

The DUT did not send current SourceToken for the audio source configuration from the GetAudioSourceConfigurationsResponse message in InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message.

#### **4.6.4 PROFILES AND AUDIO SOURCE CONFIGURATION OPTIONS CONSISTENCY**

**Test Label:** Media Service DUT GetProfiles Command and GetAudioSourceConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-3-2-4

**ONVIF Core Specification Coverage:** Get media profiles, Get audio source configuration options,

**Command Under Test:** GetProfiles, GetAudioSourceConfigurationOptions

**WSDL Reference:** media.wsdl

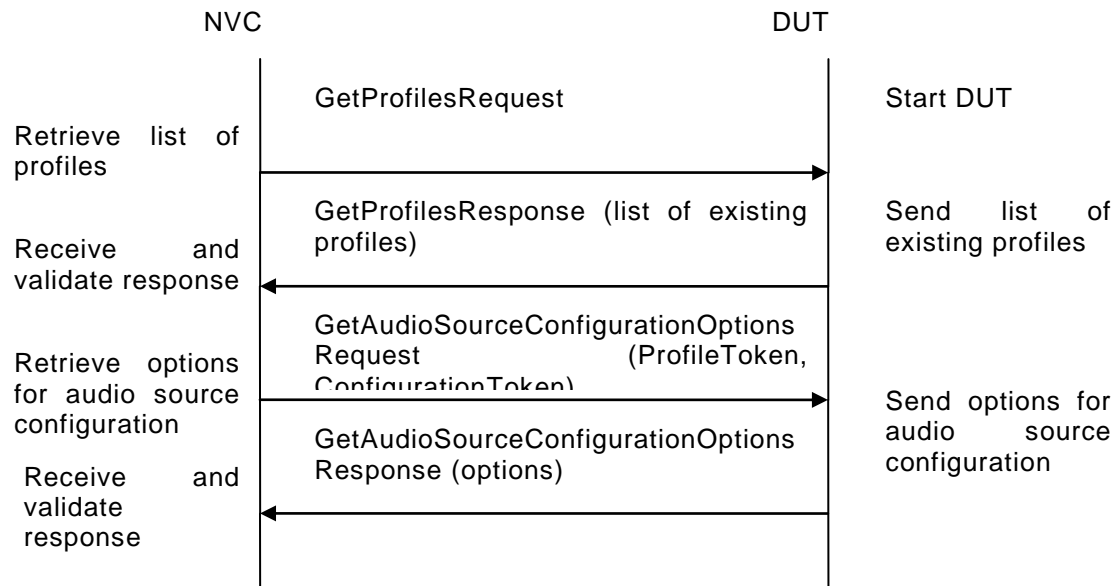
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetProfiles command and GetAudioSourceConfigurationOptions command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetProfilesRequest message to retrieve list of existing profiles from device.
4. The DUT sends GetProfilesResponse message.
5. NVC invokes GetAudioSourceConfigurationOptionsRequest (**ProfileToken**, **ConfigurationToken**) message to retrieve available audio source configuration options for profile and audio source configuration from device.
6. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
7. Check that InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message contains only unique tokens.
8. Check that SourceToken for the audio source configuration from GetProfilesResponse message exists in InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message.
9. Repeat steps 5-8 for each audio source configuration from the GetProfilesResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetProfilesResponse message.



The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationOptionsResponse message.

The DUT did not send valid GetAudioSourceConfigurationOptionsResponse message.

The DUT sent InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message that contains not unique tokens.

The DUT did not send current SourceToken for the audio source configuration from GetProfilesResponse message in InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message.

#### 4.6.5 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCES CONSISTENCY

**Test Label:** Media Service DUT GetAudioSourceConfigurations Command and GetAudioSources Command Consistency Validation.

**Test Case ID:** MEDIA-3-2-5

**ONVIF Core Specification Coverage:** Get audio source configurations, Get audio sources

**Command Under Test:** GetAudioSourceConfigurations, GetAudioSources

**WSDL Reference:** media.wsdl

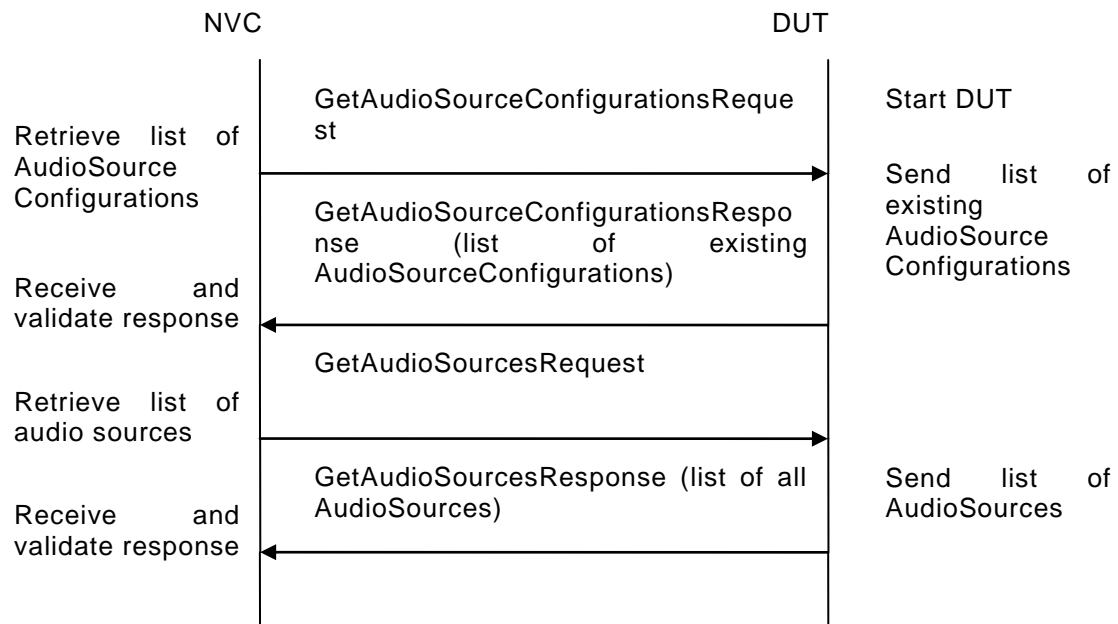
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetAudioSourceConfigurations command and GetAudioSources command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. NVC invokes GetAudioSourcesRequest message to retrieve list of Audio Sources from the DUT.
6. The DUT sends GetAudioSourcesResponse message.
7. Check that each AudioSource from GetAudioSourcesResponse message has unique token.
8. Check that every AudioSourceConfiguration.SourceToken from GetAudioSourceConfigurationsResponse message exists in GetAudioSourcesResponse message (AudioSource.token).

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourcesResponse message.

The DUT did not send valid GetAudioSourcesResponse message.

The DUT return two or more AudioSources in GetAudioSourcesResponse message with the same VideoSourceToken.

The DUT returned the GetAudioSourceConfigurationsResponse message with AudioSources that were not included in the GetAudioSourceConfigurationsResponse message.

#### **4.6.6 AUDIO SOURCE CONFIGURATION USE COUNT (CURRENT STATE)**

**Test Label:** Media Service DUT Audio Source Configuration Use Count Validation.

**Test Case ID:** MEDIA-3-2-6

**ONVIF Core Specification Coverage:** Get media profiles, Get audio source configurations, Get audio source configuration

**Command Under Test:** GetProfiles, GetAudioSourceConfigurations, GetAudioSourceConfiguration

**WSDL Reference:** media.wsdl

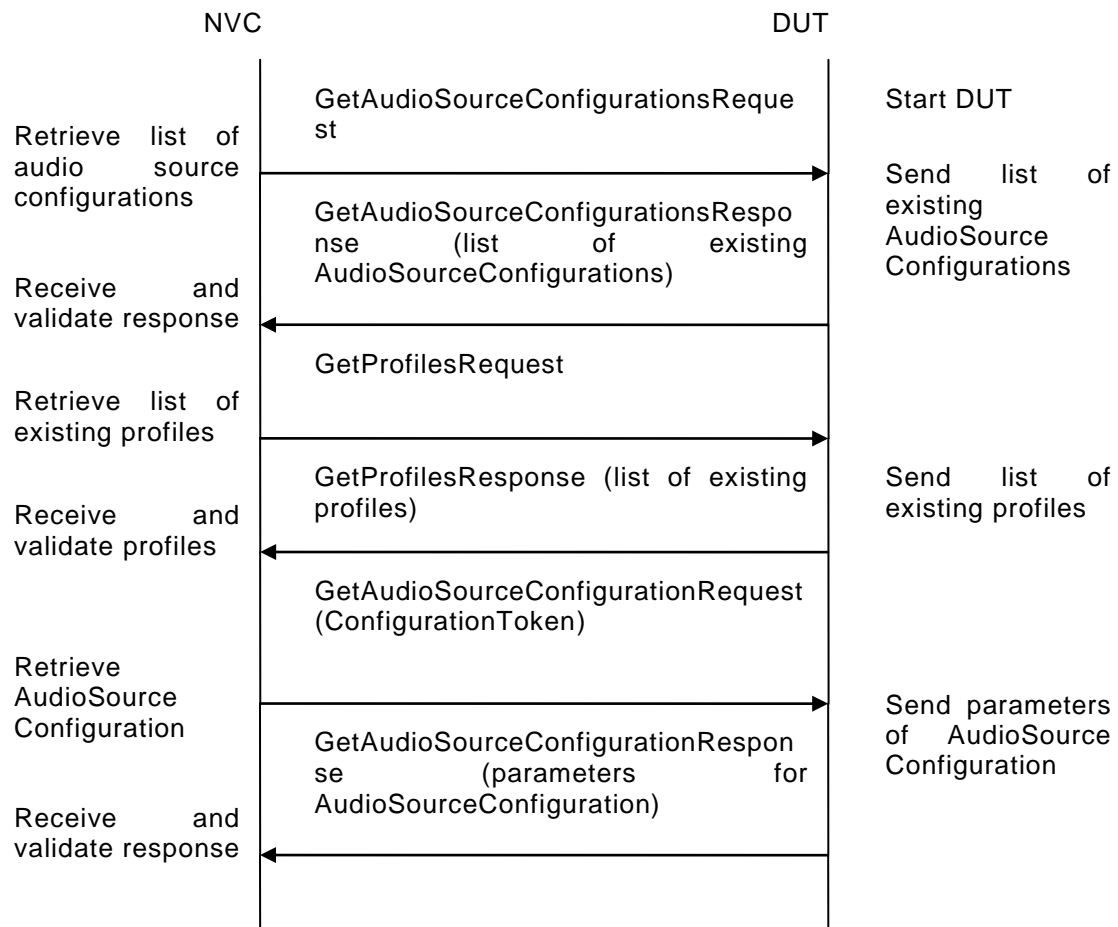
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Source Configuration use count.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. Profile with AudioSourceConfiguration exists. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. NVC invokes GetProfilesRequest message to retrieve list of profiles and their audio source configurations from device.
6. The DUT sends GetProfilesResponse message.
7. Check the **UseCount=usecount1** value for the first AudioSourceConfiguration (ASC1) in the list on step 3.
8. Check that there are not more than **usecount1** profiles with this AudioSourceConfiguration in the list from step 6.
9. Check that **UseCount** value in GetProfilesResponse for every occurrence of this AudioSourceConfiguration is **usecount1**.

10. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=the first AudioSourceConfiguration from list on step 3**) message to retrieve audio source configuration parameters.
11. The DUT sends GetAudioSourceConfigurationResponse message.
12. Check the UseCount value in GetAudioSourceConfigurationResponse (**Usecount=usecount1**).
13. Repeat steps 7-10 for all other AudioSourceConfigurations from the list on step 4.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with AudioSourceConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetAudioSourceConfigurationsResponse and GetAudioSourceConfigurationResponse messages.

#### 4.6.7 AUDIO SOURCE CONFIGURATION USE COUNT (ADD SAME AUDIO SOURCE CONFIGURATION TO PROFILE TWICE)

**Test Label:** Media Service DUT Audio Source Configuration Use Count Validation after Adding of Same Audio Source Configuration to Profile Twice.

**Test Case ID:** MEDIA-3-2-7

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile, Add audio source configuration to a profile, Get audio source configurations, Get audio source configuration

**Command Under Test:** GetAudioSourceConfigurations, GetAudioSourceConfiguration, AddAudioSourceConfiguration

**WSDL Reference:** media.wsdl

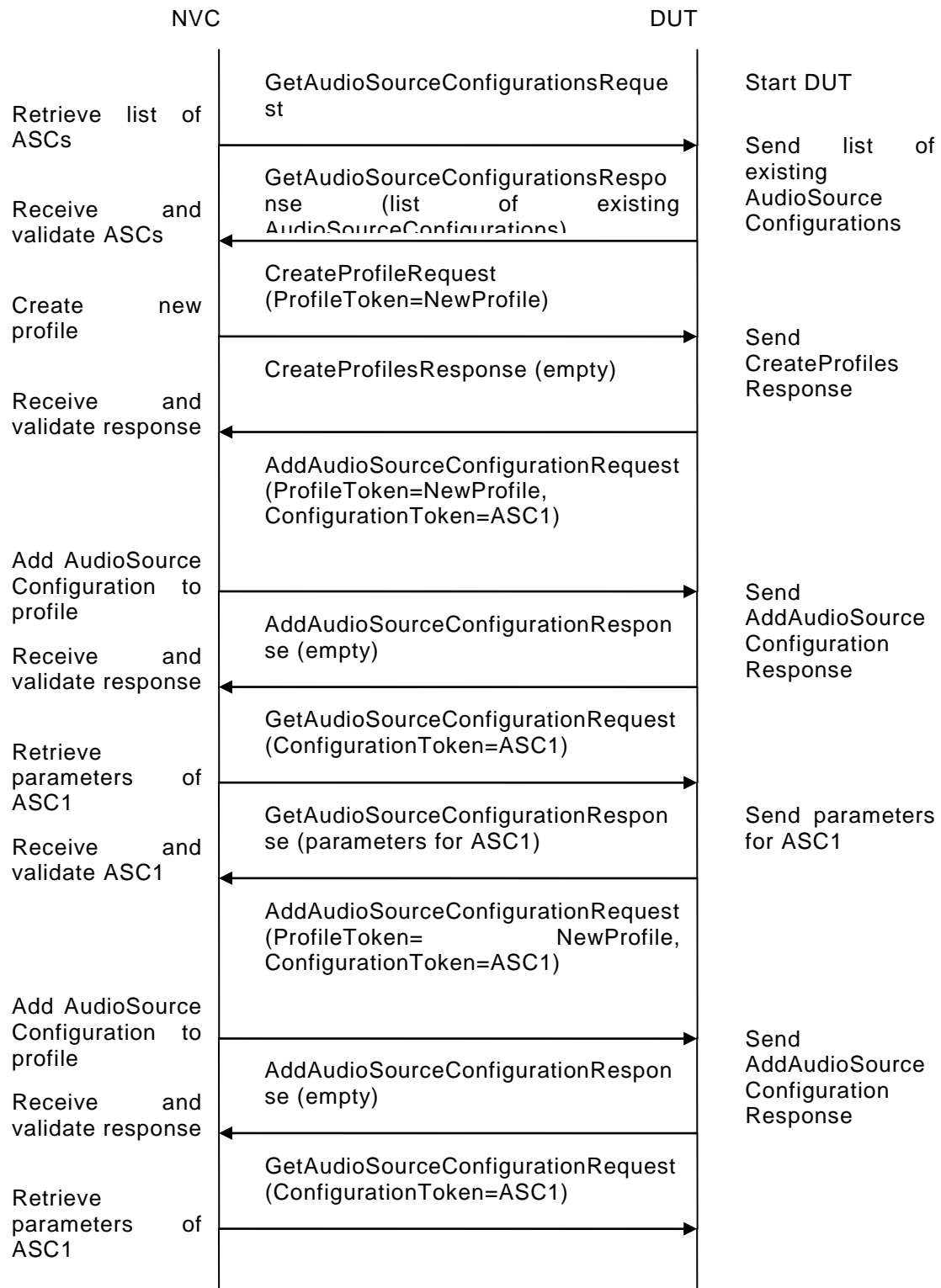
**Requirement Level:** MUST IF SUPPORTED (Audio)

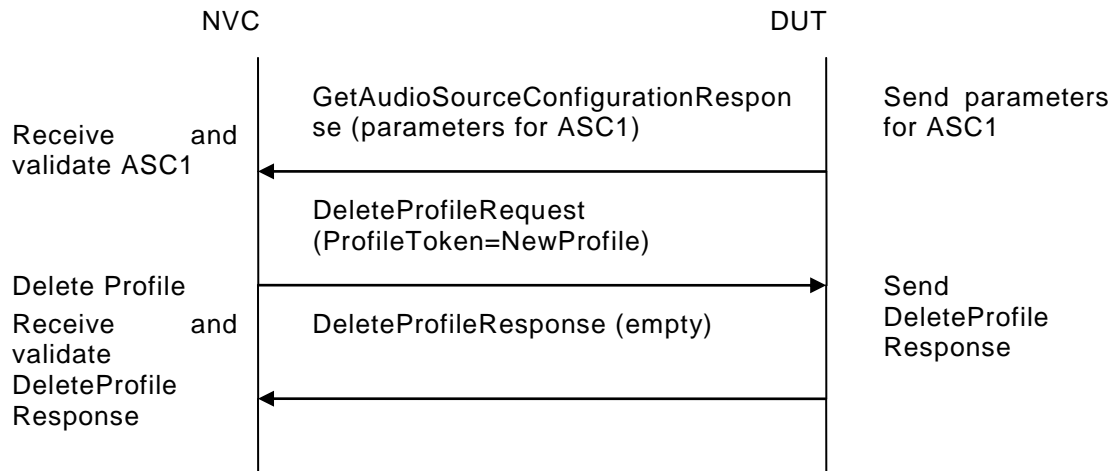
**Test Propose:** To check Audio Source Configuration use count after adding same audio source configuration to profile twice.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles Command

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. NVC invokes CreateProfileRequest (**Name=NewName**, **Token=NewProfile**) message to create profile.
6. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
NVC invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken=profile1**) message to remove AudioEncoderConfiguration and AudioSourceConfiguration from profile1  
The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message  
Execute steps 8 - 17.  
If there is no profile with "fixed" attribute value "false" in profiles list, end test.
7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio source from the list on step 4 (ASC1)**, **ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
9. The DUT sends AddAudioSourceConfigurationResponse message.
10. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
11. The DUT sends GetAudioSourceConfigurationResponse message.



12. Check the UseCount value in GetAudioSourceConfigurationResponse message (**UseCount=usecount1+1**, usecount1 is value of UseCount for ASC1 from the list on step 4). If test step 6 is executed, the usecount1 could be reduced by 1.
13. NVC invokes AddAudioSourceConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=ASC1**) message to replace audio source configuration in profile.
14. The DUT sends AddAudioSourceConfigurationResponse message.
15. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
16. The DUT sends GetAudioSourceConfigurationResponse message.
17. Check that **UseCount=usecount1+1**, in GetAudioSourceConfigurationResponse message.
18. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio source configuration.
19. The DUT sends DeleteProfileResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioSourceConfiguration to one more profile.

UseCount value is changed after repeated adding of the AudioSourceConfiguration in profile.

#### 4.6.8 AUDIO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO SOURCE CONFIGURATIONS IN PROFILE)

**Test Label:** Media Service DUT Audio Source Configuration Use Count Validation after Adding Different Audio Source Configurations to the Profile.

**Test Case ID:** MEDIA-3-2-8

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile, Add audio source configuration to a profile, Get audio source configurations, Get audio source configuration

**Command Under Test:** GetAudioSourceConfigurations, GetAudioSourceConfiguration, AddAudioSourceConfiguration

**WSDL Reference:** media.wsdl

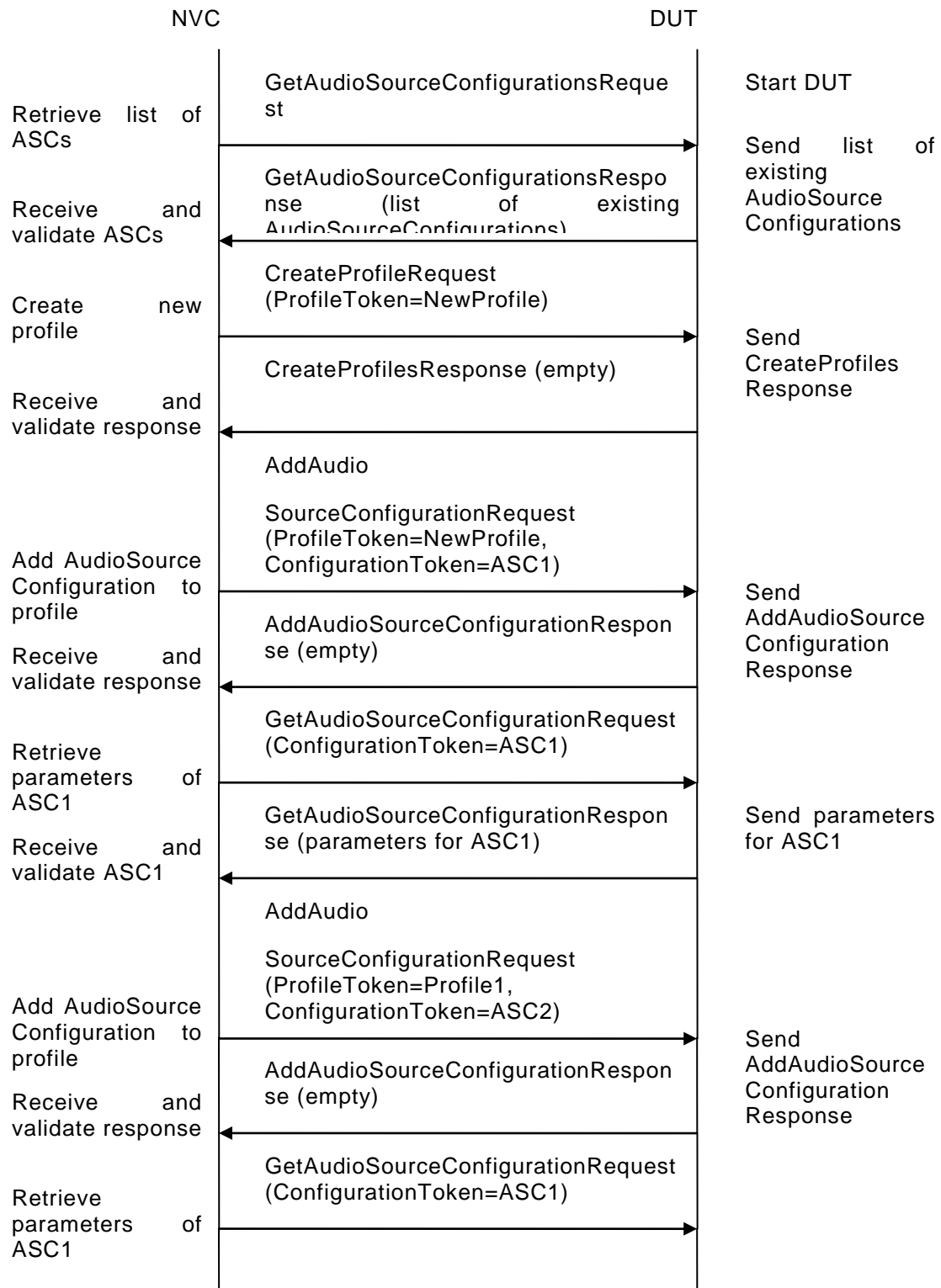
**Requirement Level:** MUST IF SUPPORTED (Audio)

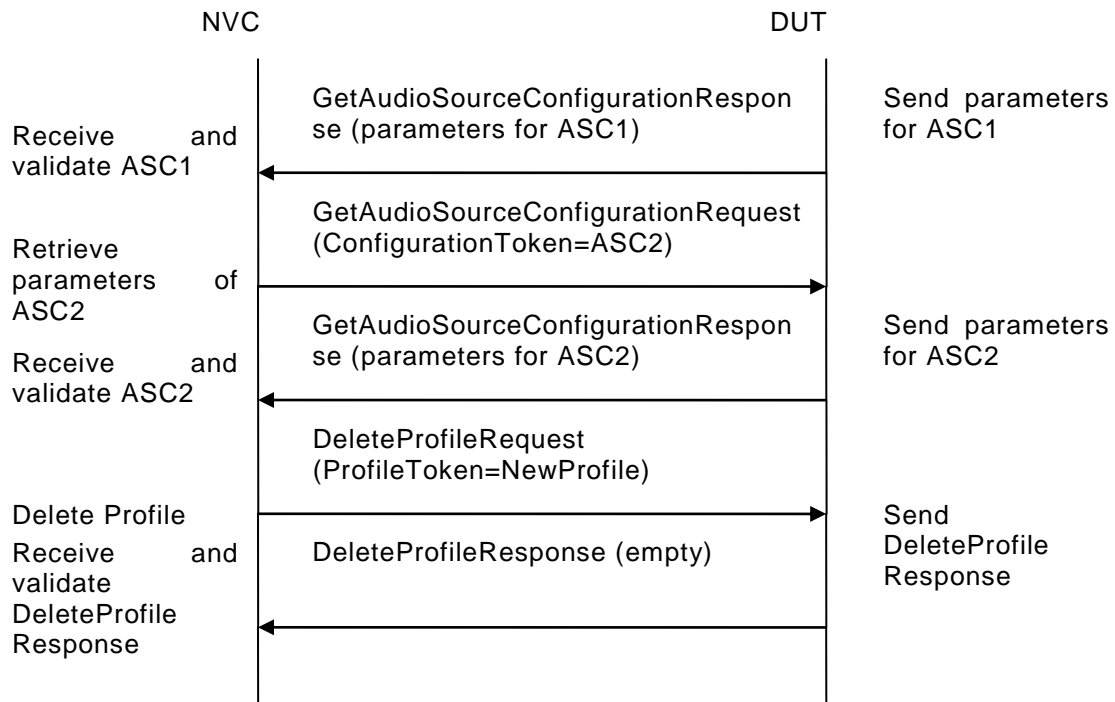
**Test Propose:** To check Audio Source Configuration use count after adding different audio source configurations to profile.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. There are at least two AudioSourceConfigurations. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. If there is only one AudioSourceConfiguration in the GetAudioSourceConfigurationsResponse message go to the next test.
6. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
7. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTPProfiles**),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

NVC invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken=profile1**) message to remove AudioSourceConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message

Execute steps 9-21

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

8. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
9. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio source from the list on step 4 (ASC1), ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
10. The DUT sends AddAudioSourceConfigurationResponse message.
11. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
12. The DUT sends GetAudioSourceConfigurationResponse message.
13. Check the UseCount value in GetAudioSourceConfigurationResponse (**UseCount=usecount1+1**, usecount1 is value of UseCount for ASC1 from the list on step 4). If test step 7 is executed, the usecount1 value could be reduced by 1.
14. NVC invokes AddAudioSourceConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=ASC2**, where ASC2 is other ASC from list on step 4) message to replace audio source configuration in profile.
15. The DUT sends AddAudioSourceConfigurationResponse message.
16. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
17. The DUT sends GetAudioSourceConfigurationResponse message.
18. Check that **UseCount=usecount1**, in GetAudioSourceConfigurationResponse.
19. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC2**) message to retrieve audio source configuration parameters.
20. The DUT sends GetAudioSourceConfigurationResponse message.
21. Check that **UseCount=usecount2+1**, in GetAudioSourceConfigurationResponse where usecount2 is UseCount value for ASC2 in the list on step 4.
22. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio source configuration.
23. The DUT sends DeleteProfileResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioSourceConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the AudioSourceConfiguration in profile for removed AudioSourceConfiguration and UseCount value is not increased by 1 after replacing of the AudioSourceConfiguration in profile for new AudioSourceConfiguration.

#### 4.6.9 AUDIO SOURCE CONFIGURATION USE COUNT (REMOVE AUDIO SOURCE CONFIGURATION)

**Test Label:** Media Service DUT Audio Source Configuration Use Count Validation after Removing of Audio Source Configuration from Profile

**Test Case ID:** MEDIA-3-2-9

**ONVIF Core Specification Coverage:** Remove audio source configuration from a profile, Get audio source configurations, Get audio source configuration

**Command Under Test:** GetProfiles, GetAudioSourceConfigurations, GetAudioSourceConfiguration, RemoveAudioSourceConfiguration

**WSDL Reference:** media.wsdl

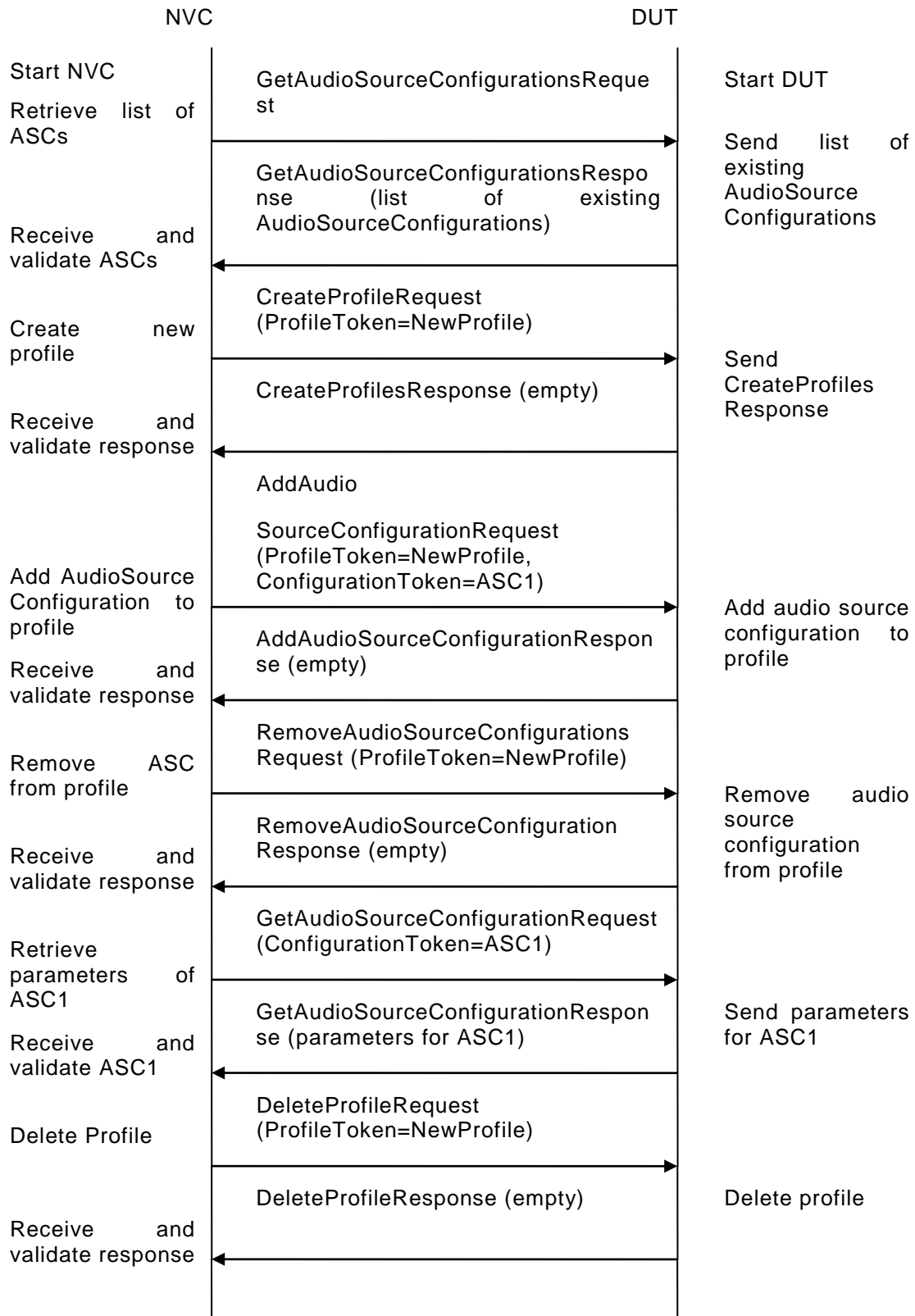
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Source Configuration use count after removing audio source configuration from profile.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles are received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
6. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
If there is profile with “fixed” attribute value “false” in profiles list (**profile1**)  
  
NVC invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken=profile1**) message to remove AudioSourceConfiguration from profile1  
  
The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message  
  
Execute steps 8-14  
  
If there is no profile with “fixed” attribute value “false” in profiles list, end test.
7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio source from the list on step 4 (ASC1), ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
9. The DUT sends AddAudioSourceConfigurationResponse message.
10. NVC invokes RemoveAudioSourceConfigurationRequest (**ProfileToken=NewProfile**) message to remove audio source configuration from profile1.
11. The DUT sends RemoveAudioSourceConfigurationResponse message.
12. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
13. The DUT sends GetAudioSourceConfigurationResponse message.
14. Check that **UseCount=usecount1**, in GetAudioSourceConfigurationResponse message where usecount1 is UseCount value for ASC1 in the list on step 4. If test step 6 is executed, the usecount1 value could be reduced by 1.
15. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio source configuration.
16. The DUT sends DeleteProfileResponse message.

**Test Result:**

**PASS –**



DUT passes all assertions.

#### **FAIL –**

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send RemoveAudioSourceConfigurationResponse message.

The DUT did not send valid RemoveAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not decreased by 1 after removing of the AudioSourceConfiguration from profile.

#### **4.6.10 AUDIO SOURCE CONFIGURATION USE COUNT (PROFILE DELETION WITH AUDIO SOURCE CONFIGURATION)**

**Test Label:** Media Service DUT Audio Source Configuration Use Count Validation after Deletion of Profile with Audio Source Configuration in it.

**Test Case ID:** MEDIA-3-2-10

**ONVIF Core Specification Coverage:** Get media profiles, Get audio source configurations, Get audio source configuration, Delete media profile

**Command Under Test:** GetProfiles, GetProfiles, GetAudioSourceConfigurations, GetAudioSourceConfiguration, SetAudioSourceConfiguration, DeleteProfile

**WSDL Reference:** media.wsdl

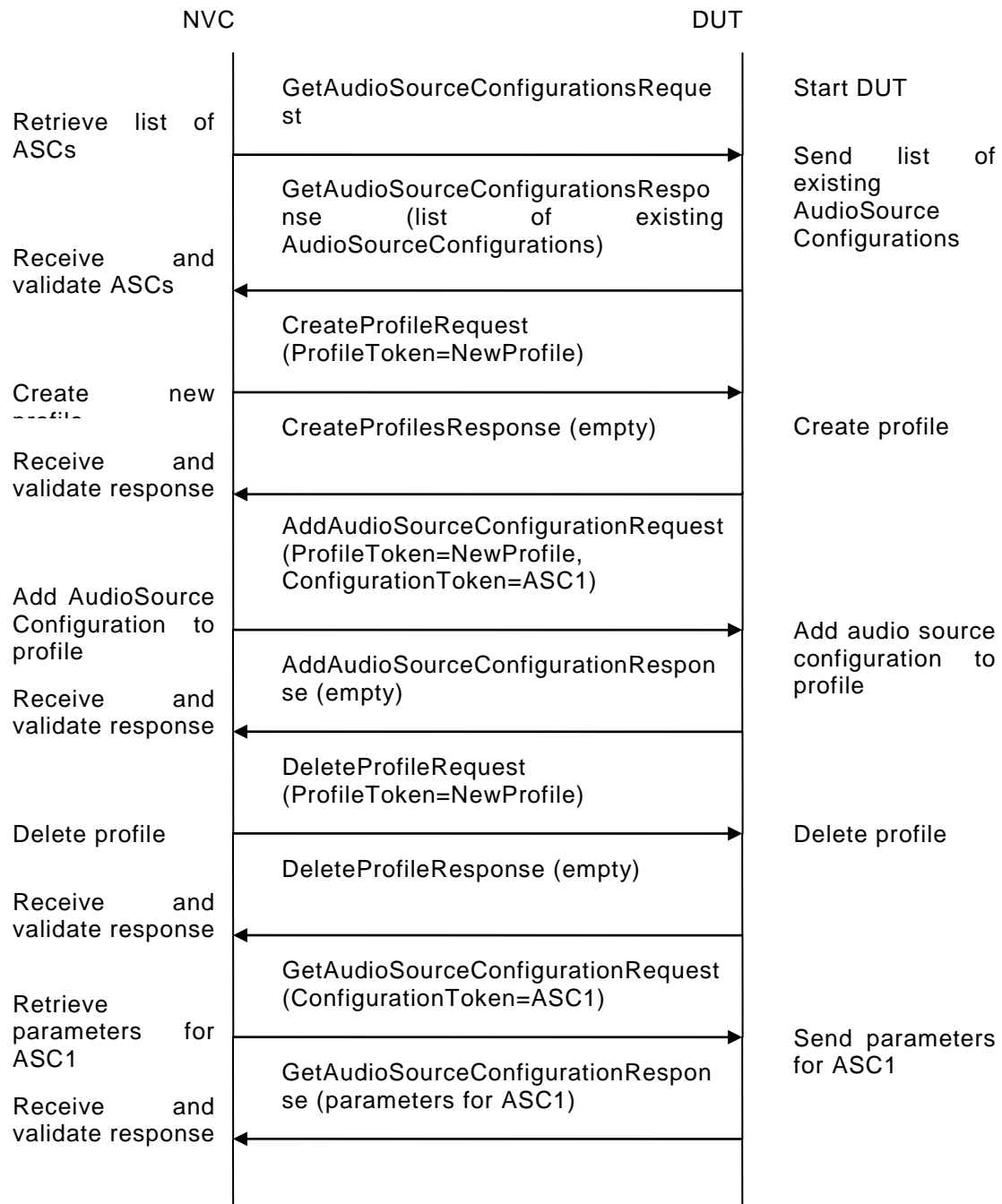
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Source Configuration use count after deletion of profile with audio source configuration in it.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command/ List of profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.

5. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
6. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
NVC invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken=profile1**) message to remove AudioSourceConfiguration from profile1  
The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message  
Execute steps 8-14  
If there is no profile with "fixed" attribute value "false" in profiles list, end test.
7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
8. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio source from the list on step 4 (ASC1), ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
9. The DUT sends AddAudioSourceConfigurationResponse message.
10. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio source configuration.
11. The DUT sends DeleteProfileResponse message.
12. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
13. The DUT sends GetAudioSourceConfigurationResponse message.
14. Check that **UseCount=usecount1**, in GetAudioSourceConfigurationResponse where **usecount1** is UseCount value for ASC1 in the list on step 4. If test step 6 is executed, the usecount1 value could be reduced by 1.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

The DUT did not send valid DeleteProfileResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the AudioSourceConfiguration.

#### **4.6.11 AUDIO SOURCE CONFIGURATION USE COUNT (SET AUDIO SOURCE CONFIGURATION)**

**Test Label:** Media Service DUT Audio Source Configuration Use Count Validation after Changing of Audio Source Configuration.

**Test Case ID:** MEDIA-3-2-11

**ONVIF Core Specification Coverage:** Get audio source configurations, Get audio source configuration, Modify an audio source configuration.

**Command Under Test:** GetAudioSourceConfigurations, GetAudioSourceConfiguration, SetAudioSourceConfiguration

**WSDL Reference:** media.wsdl

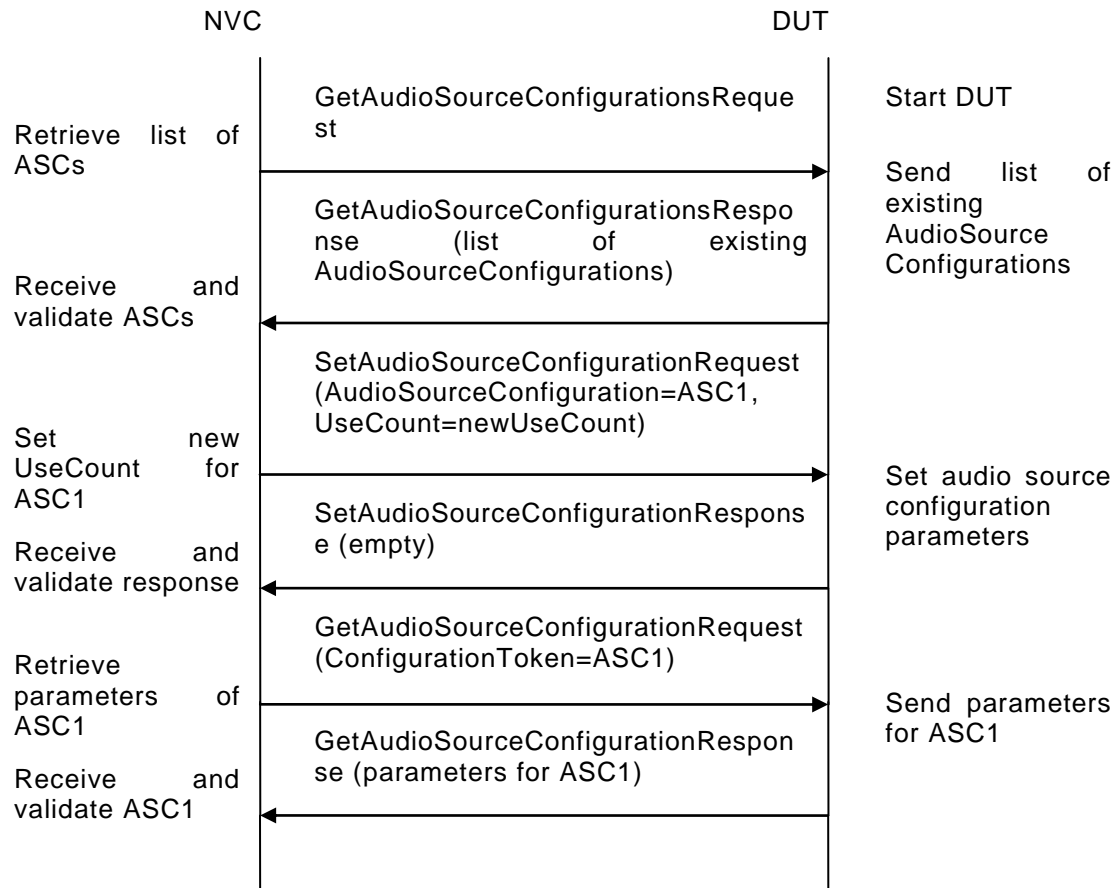
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Source Configuration use count after changing of audio source configuration parameters.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
4. The DUT sends GetAudioSourceConfigurationsResponse message.
5. NVC invokes SetAudioSourceConfigurationRequest (**token=the first Audio Source Configuration from the list on step 4(ASC1), UseCount=NewUseCount**) message to set parameters for audio source configuration.
6. The DUT sends SetAudioSourceConfigurationResponse message.
7. NVC invokes GetAudioSourceConfigurationRequest (**ConfigurationToken=ASC1**) message to retrieve audio source configuration parameters.
8. The DUT sends GetAudioSourceConfigurationResponse message.
9. Check that **UseCount =usecount1**, in GetAudioSourceConfigurationResponse where usecount1 is UseCount value for ASC1 in the list on step 4.

#### Test Result:

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send SetAudioSourceConfigurationResponse message.

The DUT did not send valid SetAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value was changed after trying to set UseCount value.

## **4.7 Audio Encoder Configuration**

### **4.7.1 AUDIO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY**

**Test Label:** Media Service DUT Audio Encoder Configurations and Profiles Consistency Validation.

**Test Case ID:** MEDIA-3-3-1

**ONVIF Core Specification Coverage:** Get media profiles, Get audio encoder configurations

**Command Under Test:** GetProfiles, GetAudioEncoderConfigurations

**WSDL Reference:** media.wsdl

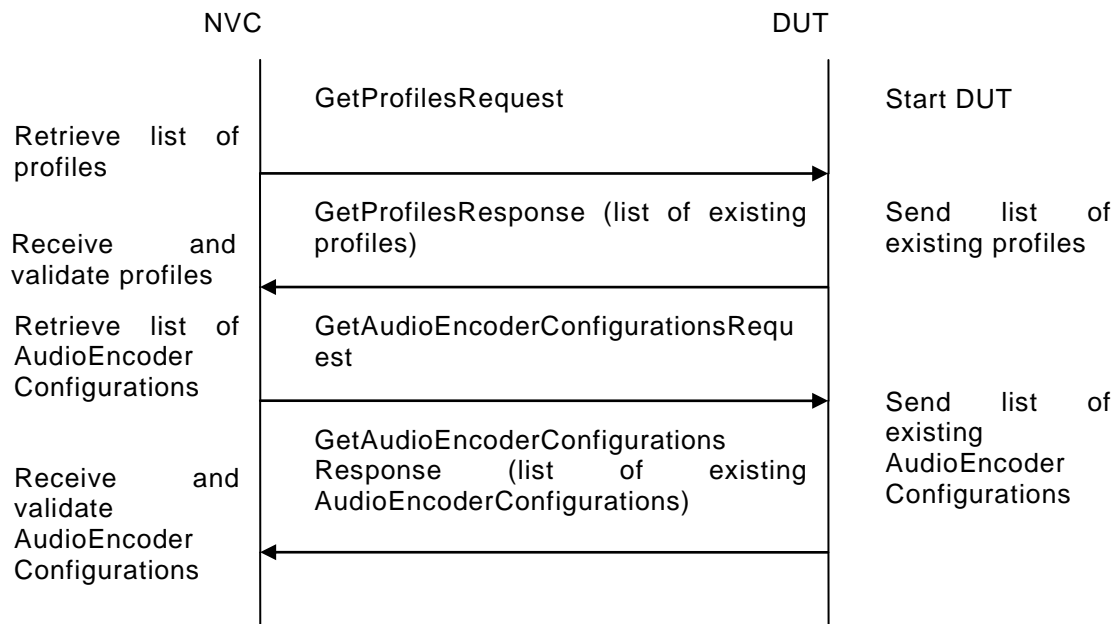
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that Audio Encoder Configurations and Audio Encoder Profiles are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes **GetProfilesRequest** message to retrieve list of profiles and their audio encoder configurations.
4. The DUT sends **GetProfilesResponse** message.
5. NVC invokes **GetAudioEncoderConfigurationsRequest** message to retrieve list of Audio Source Configurations from device.
6. The DUT sends **GetAudioEncoderConfigurationsResponse** message.
7. Check that each **GetAudioEncoderConfiguration** from **GetAudioEncoderConfigurationsResponse** message has unique token.
8. Check that each **AudioEncoderConfiguration** from the **GetProfilesResponse** message are included in the **GetAudioEncoderConfigurationsResponse** message.
9. Check that **AudioEncoderConfiguration** parameters are same in the **GetProfilesResponse** message and in the **GetAudioEncoderConfigurationsResponse** message for corresponding **AudioEncoderConfiguration**.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT return two or more AudioEncoderConfigurations in GetAudioEncoderConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with AudioEncoderConfigurations that were not included in the GetAudioEncoderConfigurationsResponse message.

The DUT returned different parameters list and parameters values in the GetAudioEncoderConfigurationsResponse message and in the GetProfilesResponse message for the same AudioEncoderConfiguration.

#### 4.7.2 AUDIO ENCODER CONFIGURATIONS AND AUDIO ENCODER CONFIGURATION

##### CONSISTENCY

**Test Label:** Media Service DUT GetAudioEncoderConfigurations Command and GetAudioEncoderConfiguration Command Consistency Validation.

**Test Case ID:** MEDIA-3-3-2

**ONVIF Core Specification Coverage:** Get audio encoder configurations, Get audio encoder configuration

**Command Under Test:** GetAudioEncoderConfigurations, GetAudioEncoderConfiguration

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF SUPPORTED (Audio)

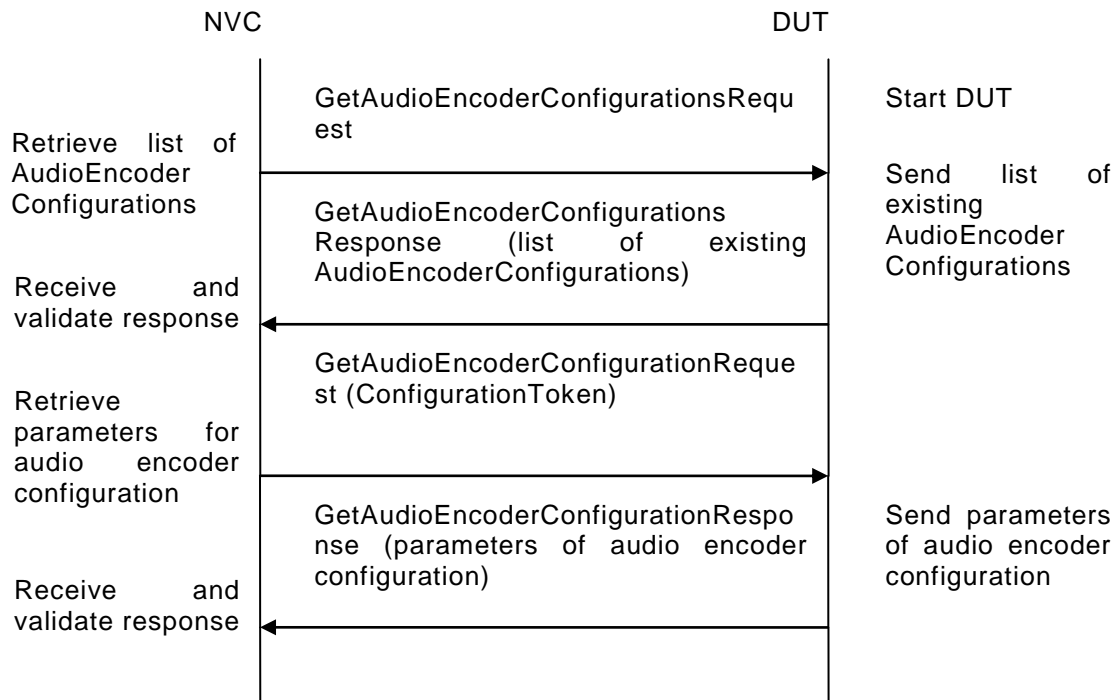
**Test Propose:** To check that GetAudioEncoderConfigurations command and GetAudioEncoderConfiguration command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioEncoderConfigurationsRequest` message to retrieve list of Audio Source Configurations from device.
4. The DUT sends `GetAudioEncoderConfigurationsResponse` message.
5. NVC invokes `GetAudioEncoderConfigurationRequest (ConfigurationToken)` message to retrieve parameters of audio encoder configuration from the DUT.
6. The DUT sends `GetAudioEncoderConfigurationResponse` message.
7. Verify that all parameters for audio encoder configuration from the `GetAudioEncoderConfigurationsResponse` message and from the `GetAudioEncoderConfigurationResponse` message are the same.
8. Repeat steps 5-7 for each Audio Encoder Configurations from the `GetAudioEncoderConfigurationsResponse` message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT did not send equal parameters for one or more AudioEncoderConfiguration in the GetAudioEncoderConfigurationResponse message and in the GetAudioEncoderConfigurationsResponse message.

#### 4.7.3 AUDIO ENCODER CONFIGURATIONS AND AUDIO ENCODER CONFIGURATION OPTIONS

##### CONSISTENCY

**Test Label:** Media Service DUT GetAudioEncoderConfigurations Command and GetAudioEncoderConfigurationOptions Command Consistency Validation.

**Test Case ID:** MEDIA-3-3-3

**ONVIF Core Specification Coverage:** Get audio encoder configurations, Get audio encoder configuration options

**Command Under Test:** GetAudioEncoderConfigurations, GetAudioEncoderConfigurationOptions

**WSDL Reference:** media.wsdl

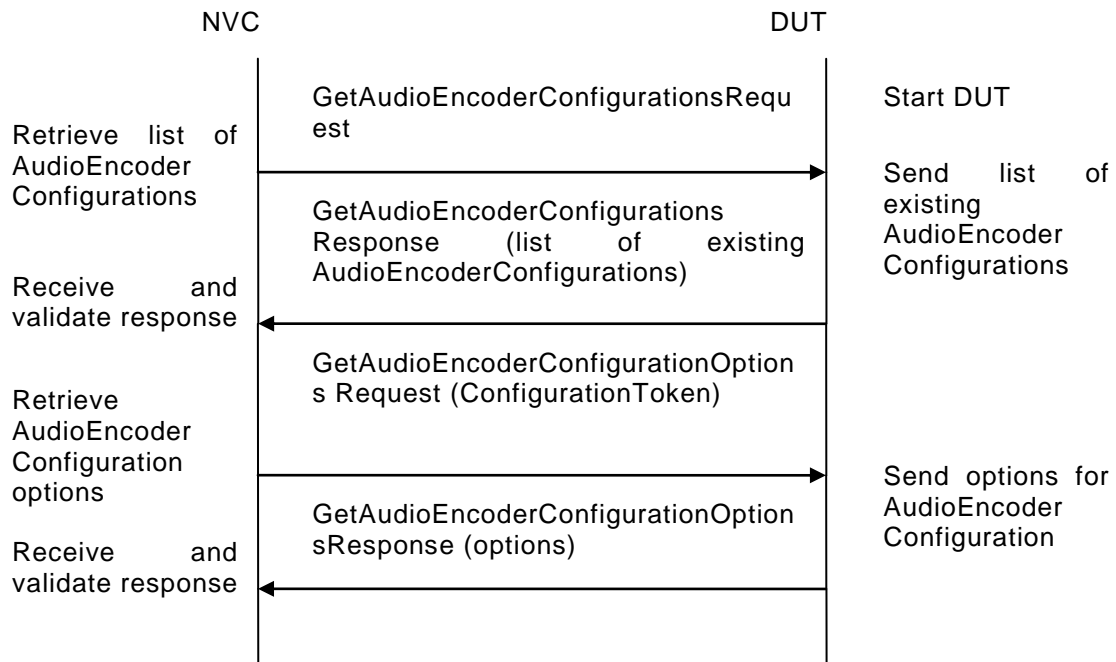
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetAudioEncoderConfiguration command and GetAudioEncoderConfigurationOptions command are consistent.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioEncoderConfigurationsRequest` message to retrieve list of Audio Encoder Configurations from device.
4. The DUT sends `GetAudioEncoderConfigurationsResponse` message.
5. NVC invokes `GetAudioEncoderConfigurationOptionsRequest` (**ConfigurationToken**) message to retrieve available audio encoder configuration options from device.
6. The DUT sends `GetAudioEncoderConfigurationOptionsResponse` message.
7. Check that Encoding value for audio encoder configuration exists in Encoding parameter of one of the Option sections from list in the `GetAudioEncoderConfigurationOptionsResponse` message, and Bitrate and SampleRate exist in BitrateList and SampleRateList for this encoding in the `GetAudioEncoderConfigurationOptionsResponse` message.
8. Repeat steps 5-7 for each audio encoder configuration from the `GetAudioEncoderConfigurationsResponse` message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send `GetAudioEncoderConfigurationsResponse` message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationOptionsResponse message.

The DUT sent non-available Encoding value for one or more AudioSourceConfigurations

The DUT sent not available combination of Encoding and Bitrate or Encoding and SampleRate for one or more AudioEncoderConfigurations.

#### 4.7.4 PROFILES AND AUDIO ENCODER CONFIGURATION OPTIONS CONSISTENCY

**Test Label:** Media Service DUT GetProfiles Command and GetAudioEncoderConfigurationOptions Consistency Validation.

**Test Case ID:** MEDIA-3-3-4

**ONVIF Core Specification Coverage:** Get media profile, Get audio encoder configuration options

**Command Under Test:** GetProfiles, GetAudioEncoderConfigurationOptions

**WSDL Reference:** media.wsdl

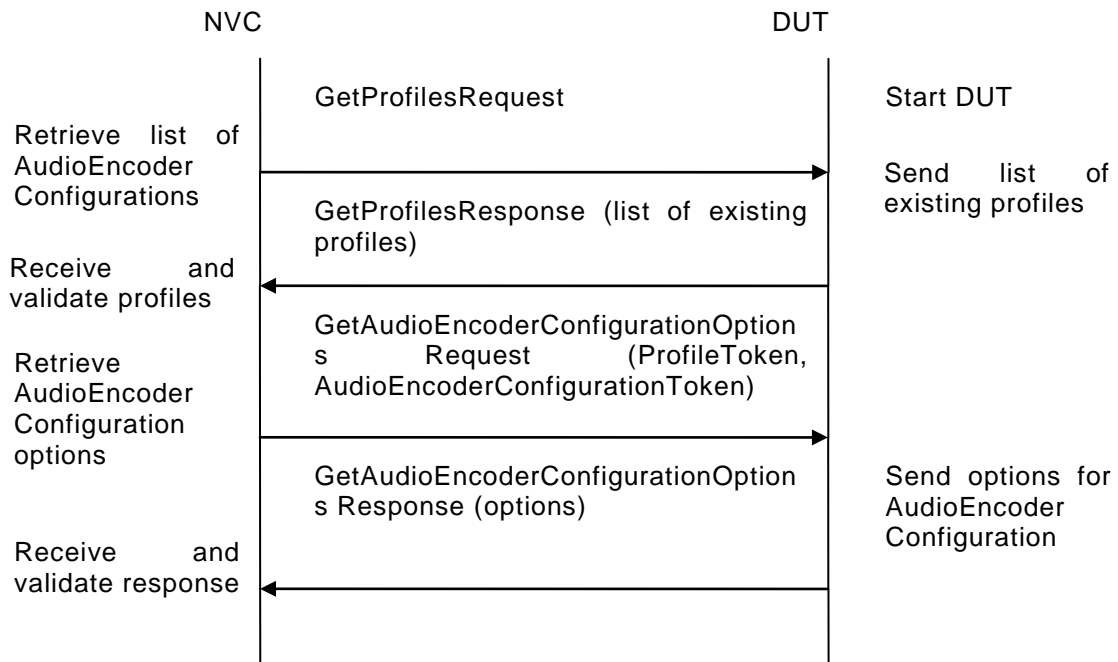
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check that GetProfiles command and GetAudioEncoder onfigurationOptions command are consistency.

**Pre-Requisite:** Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes **GetProfilesRequest** message to retrieve list of profiles from device.
4. The DUT sends **GetProfilesResponse** message.
5. NVC invokes **GetAudioEncoderConfigurationOptionsRequest** (**ProfileToken**, **ConfigurationToken**) message to retrieve **AudioEncoderConfigurationOptions** for audio encoder configuration and profile from the DUT.
6. The DUT sends **GetAudioEncoderConfigurationOptionsResponse** message.
7. Check that **Encoding** value for audio encoder configuration exists in **Encoding** parameter of one of the **Option** sections from list in the **GetAudioEncoderConfigurationOptionsResponse** message, and **Bitrate** and **SampleRate** exist in **BitrateList** and **SampleRateList** for this encoding in the **GetAudioEncoderConfigurationOptionsResponse** message.
8. Repeat steps 5-7 for each audio encoder configuration from the **GetProfilesResponse** message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send **GetProfilesResponse** message.

The DUT did not send valid **GetProfilesResponse** message.

The DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationOptionsResponse message.

The DUT sent non-available Encoding value for one or more AudioSourceConfigurations

The DUT sent not available combination of Encoding and Bitrate or Encoding and SampleRate for one or more AudioEncoderConfigurations.

#### 4.7.5 AUDIO ENCODER CONFIGURATION USE COUNT (CURRENT STATE)

**Test Label:** Media Service DUT Audio Encoder Configuration Use Count Validation.

**Test Case ID:** MEDIA-3-3-5

**ONVIF Core Specification Coverage:** Get media profiles, Get audio encoder configurations, Get audio encoder configuration

<b>Command</b>	<b>Under</b>	<b>Test:</b>	GetProfiles,	GetAudioEncoderConfigurations,
			GetAudioEncoderConfiguration	

**WSDL Reference:** media.wsdl

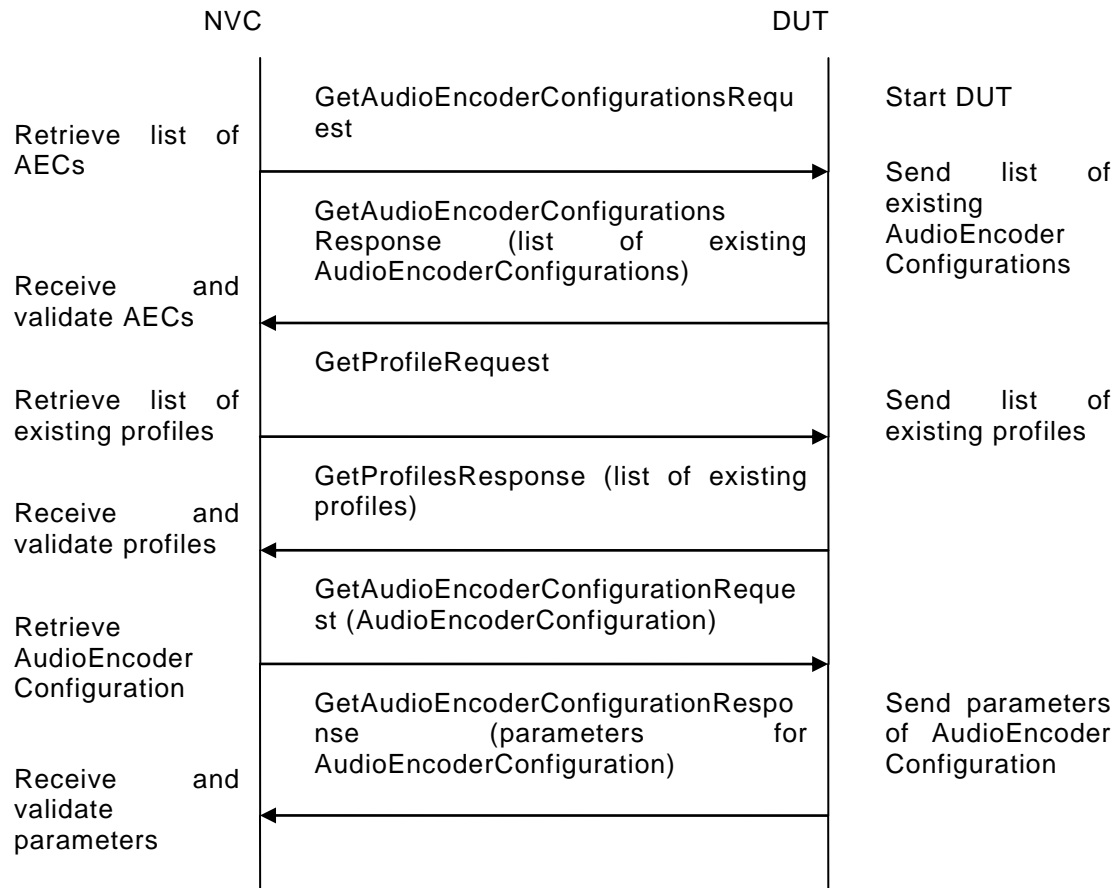
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Encoder Configuration use count.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioEncoderConfigurationsRequest` message to retrieve list of audio encoder configurations from device.
4. The DUT sends `GetAudioEncoderConfigurationsResponse` message.
5. NVC invokes `GetProfilesRequest` message to retrieve list of profiles and their audio encoder configurations from device.
6. The DUT sends `GetProfilesResponse` message.
7. Check the **UseCount=usecount1** value for the first `AudioEncoderConfiguration` (AEC1) in the list on step 3.
8. Check that there are not more than **usecount1** profiles with this `AudioEncoderConfiguration` in the list from step 6.
9. Check that **UseCount** value in `GetProfilesResponse` message for every occurrence of this `AudioEncoderConfiguration` is **usecount1**.

10. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
11. The DUT sends GetAudioEncoderConfigurationResponse message.
12. Check the UseCount value in GetAudioEncoderConfigurationResponse (**Usecount=usecount1**) message.
13. Repeat steps 7-10 for all other AudioEncoderConfigurations from the list on step 4.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with AudioEncoderConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetAudioEncoderConfigurationsResponse and GetAudioEncoderConfigurationResponse messages.

#### 4.7.6 AUDIO ENCODER CONFIGURATION USE COUNT (ADD SAME AUDIO ENCODER CONFIGURATION TO PROFILE TWICE)

**Test Label:** Media Service DUT Audio Encoder Configuration Use Count Validation after Adding of Same Audio Encoder Configuration to Profile Twice.

**Test Case ID:** MEDIA-3-3-6

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile, Add audio encoder configuration to a profile, Get audio encoder configurations, Get audio encoder configuration

**Command Under Test:** GetProfiles, GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, AddAudioEncoderConfiguration

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF SUPPORTED (Audio)

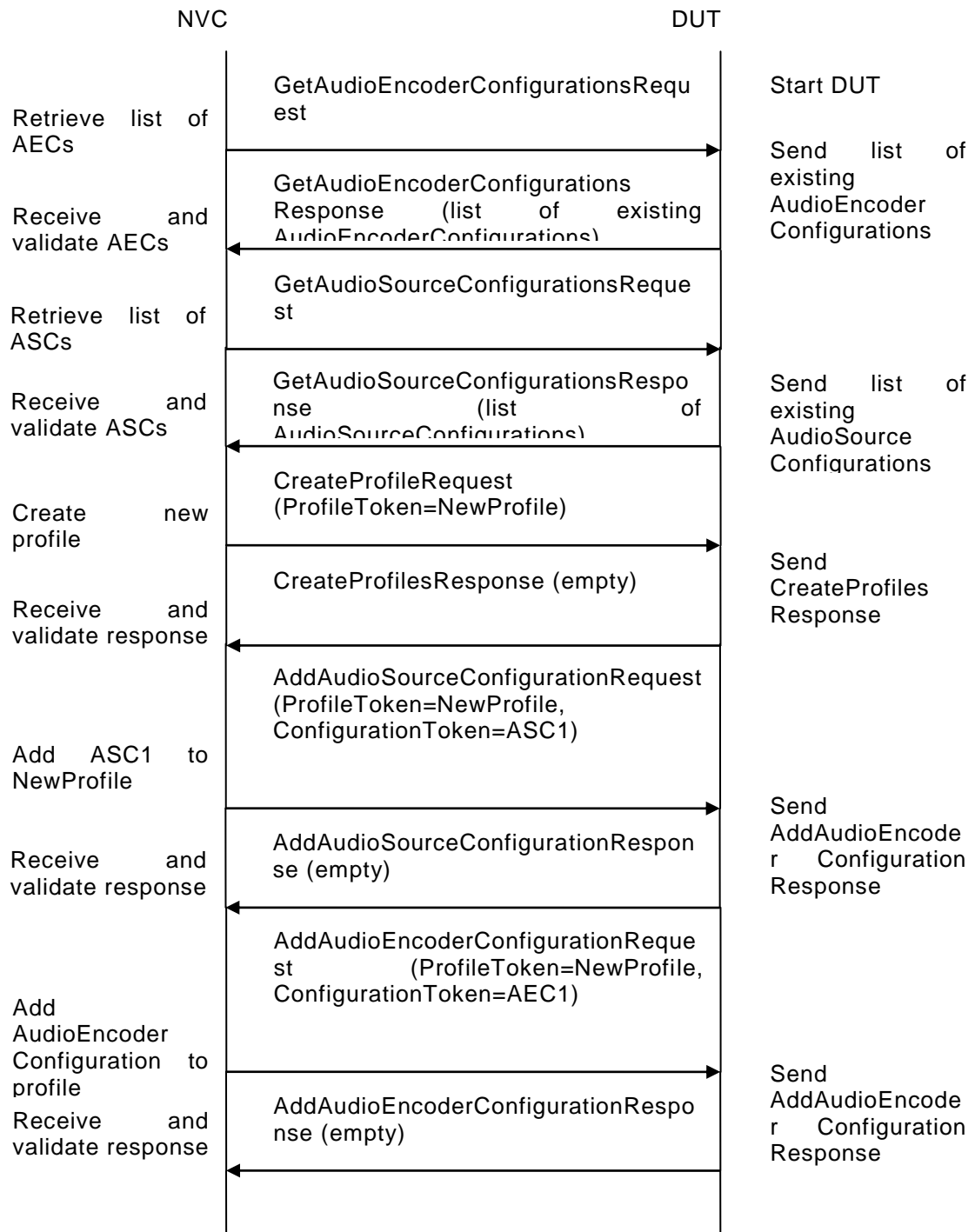


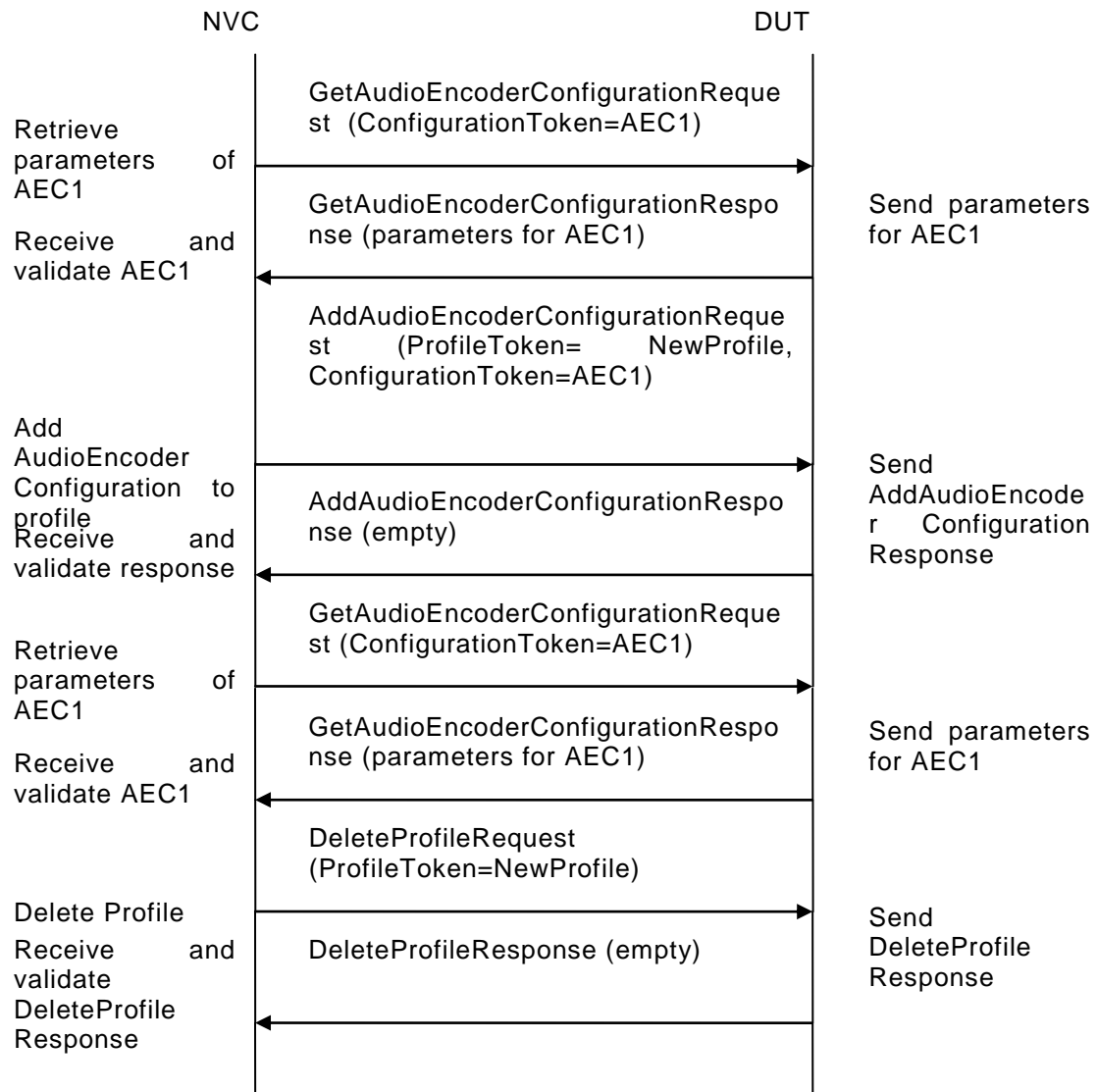
**Test Propose:** To check Audio Encoder Configuration use count after adding same audio encoder configuration to profile twice.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioEncoderConfigurationsRequest` message to retrieve list of audio encoder configurations from device.
4. The DUT sends `GetAudioEncoderConfigurationsResponse` message.
5. NVC invokes `GetAudioSourceConfigurationsRequest` message to retrieve list of audio encoder configurations from device.
6. The DUT sends `GetAudioSourceConfigurationsResponse` message.

7. NVC invokes CreateProfileRequest (**Name=NewName, Token=NewProfile**) message to create profile.
8. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
NVC invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken=profile1**) message to remove AudioEncoderConfiguration from profile1  
The DUT sends RemoveAudioEncoderConfigurationResponse message  
Execute steps 10-21  
If there is no profile with "fixed" attribute value "false" in profiles list, end test.
9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
10. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio source configuration from the list on step 6 (ASC1), ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
11. The DUT sends AddAudioSourceConfigurationResponse message.
12. NVC invokes AddAudioEncoderConfigurationRequest (**ConfigurationToken=first audio encoder from the list on step 4 (AEC1), ProfileToken=NewProfile**) message to add AudioEncoderConfiguration to the new profile.
13. The DUT sends AddAudioEncoderConfigurationResponse message.
14. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
15. The DUT sends GetAudioEncoderConfigurationResponse message.
16. Check the UseCount value in GetAudioEncoderConfigurationResponse message (**UseCount=usecount1+1**, usecount1 is value of UseCount for AEC1 from the list on step 4).  
If test step 8 is executed, the usecount1 value could be reduced by 1.
17. NVC invokes AddAudioEncoderConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=AEC1**) message to replace audio encoder configuration in profile.
18. The DUT sends AddAudioEncoderConfigurationResponse message.
19. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
20. The DUT sends GetAudioEncoderConfigurationResponse message.
21. Check that **UseCount=usecount1+1**, in GetAudioEncoderConfigurationResponse message.
22. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio encoder configuration.
23. The DUT sends DeleteProfileResponse message.

#### Test Result:

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send AddAudioEncoderConfigurationResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioEncoderConfiguration to one more profile.

UseCount value is changed after repeated adding of the AudioEncoderConfiguration in profile.

#### **4.7.7 AUDIO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO ENCODER CONFIGURATIONS IN PROFILE)**

**Test Label:** Media Service DUT Audio Encoder Configuration Use Count Validation after Adding Different Audio Encoder Configurations to the Profile.

**Test Case ID:** MEDIA-3-3-7

**ONVIF Core Specification Coverage:** Get media profiles, Get media profile, Add audio encoder configuration to a profile, Get audio encoder configurations, Get audio encoder configuration

**Command Under Test:** GetProfiles, GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, AddAudioEncoderConfiguration

**WSDL Reference:** media.wsdl

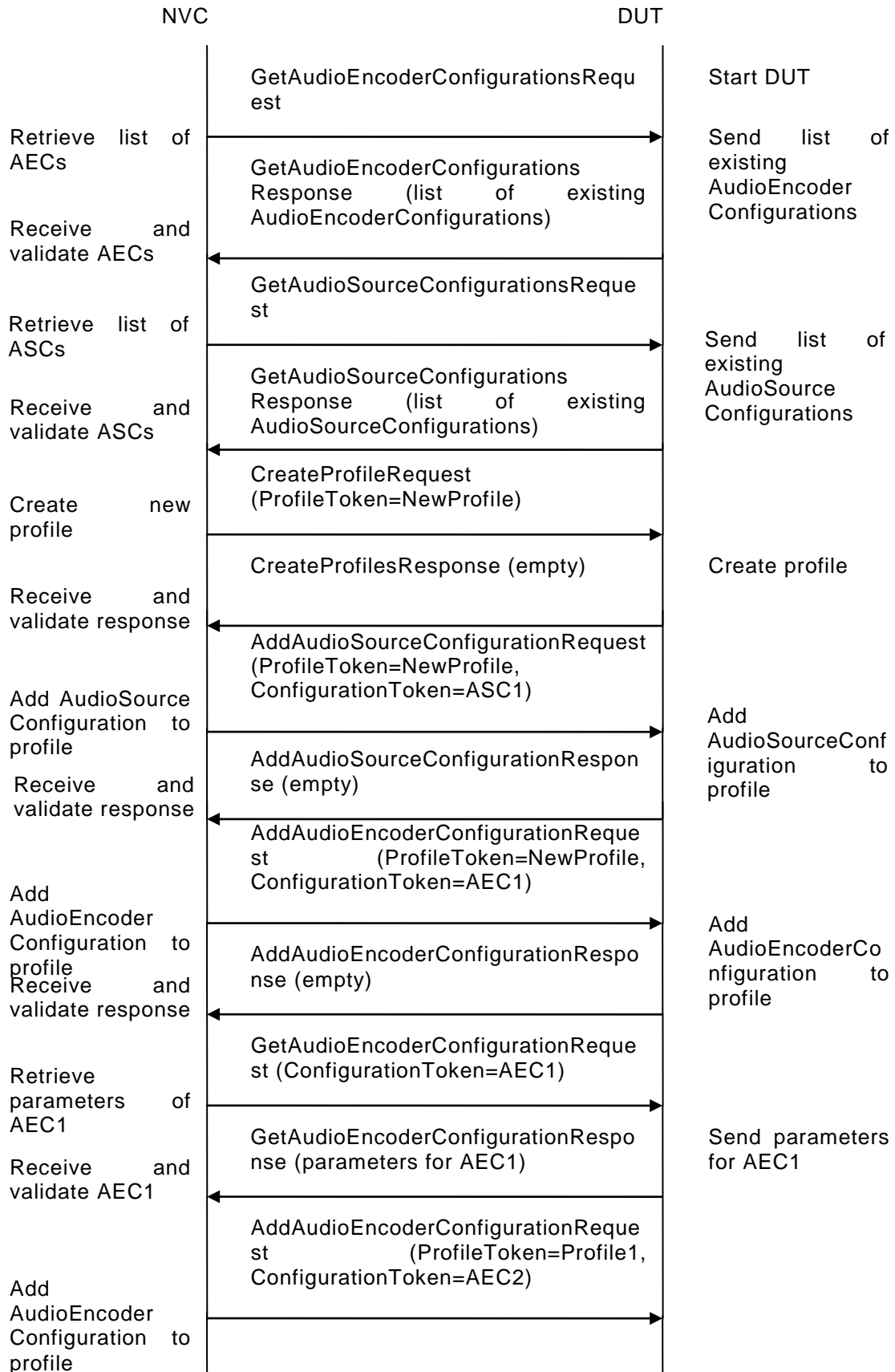
**Requirement Level:** MUST IF SUPPORTED (Audio)

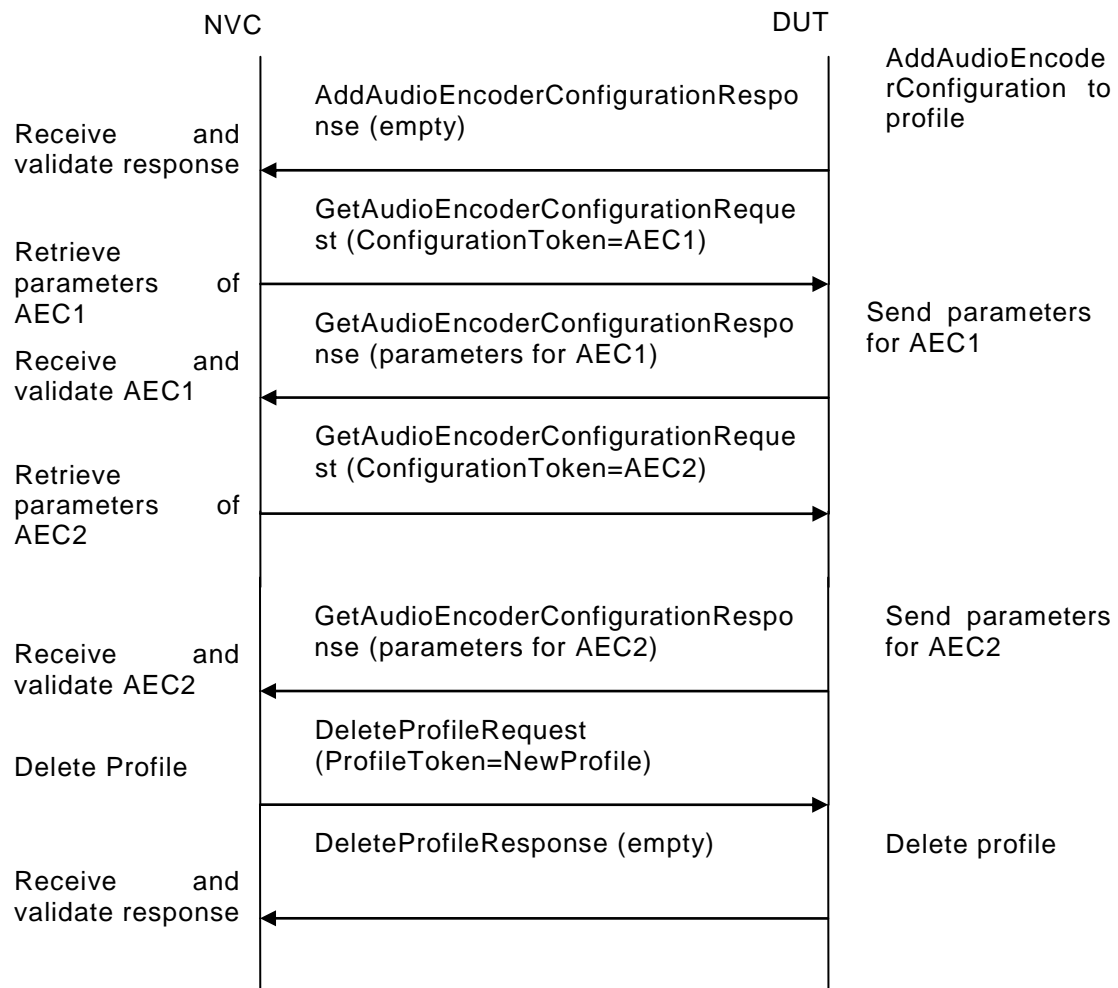
**Test Propose:** To check Audio Encoder Configuration use count after adding different audio encoder configurations to profile.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. There are at least two AudioEncoderConfigurations. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes **GetAudioEncoderConfigurationsRequest** message to retrieve list of audio encoder configurations from device.
4. The DUT sends **GetAudioEncoderConfigurationsResponse** message.
5. If there is only one **AudioEncoderConfiguration** in the **GetAudioEncoderConfigurationsResponse** message go to the next test.
6. NVC invokes **GetAudioSourceConfigurationsRequest** message to retrieve list of audio source configurations from device.
7. The DUT sends **GetAudioSourceConfigurationsResponse** message.
8. NVC invokes **CreateProfileRequest (Name=NewName, Token=NewProfile)** message to create profile.

9. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
 If there is profile with “fixed” attribute value “false” in profiles list (**profile1**)  
 NVC invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken=profile1**) message to remove AudioEncoderConfiguration from profile1  
 The DUT sends RemoveAudioEncoderConfigurationResponse message  
 Execute steps 11-25  
 If there is no profile with “fixed” attribute value “false” in profiles list, end test.
10. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
11. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio encoder from the list on step 7 (ASC1), ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
12. The DUT sends AddAudioSourceConfigurationResponse message.
13. NVC invokes AddAudioEncoderConfigurationRequest (**ConfigurationToken=first audio encoder from the list on step 4 (AEC1), ProfileToken=NewProfile**) message to add AudioEncoderConfiguration to the new profile.
14. The DUT sends AddAudioEncoderConfigurationResponse message.
15. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
16. The DUT sends GetAudioEncoderConfigurationResponse message.
17. Check the UseCount value in GetAudioEncoderConfigurationResponse (**UseCount=usecount1+1**, usecount1 is value of UseCount for AEC1 from the list on step 4). If test step 9 is executed, the usecount1 value could be reduced by 1.
18. NVC invokes AddAudioEncoderConfigurationRequest (**ProfileToken=NewProfile, ConfigurationToken=AEC2**, where AEC2 is other AEC from list on step4) message to replace audio encoder configuration in profile.
19. The DUT sends AddAudioEncoderConfigurationResponse message.
20. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
21. The DUT sends GetAudioEncoderConfigurationResponse message.
22. Check that **UseCount=usecount1**, in GetAudioEncoderConfigurationResponse.
23. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC2**) message to retrieve audio encoder configuration parameters.
24. The DUT sends GetAudioEncoderConfigurationResponse message.
25. Check that **UseCount=usecount2+1**, in GetAudioEncoderConfigurationResponse where usecount2 is UseCount value for AEC2 in the list on step 4.
26. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio encoder configuration.



27. The DUT sends DeleteProfileResponse message.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send AddAudioEncoderConfigurationResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioEncoderConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the AudioEncoderConfiguration in profile for removed AudioEncoderConfiguration and UseCount value is not increased by 1 after replacing of the AudioEncoderConfiguration in profile for new AudioEncoderConfiguration.

#### 4.7.8 AUDIO ENCODER CONFIGURATION USE COUNT (REMOVE AUDIO ENCODER CONFIGURATION)

**Test Label:** Media Service DUT Audio Encoder Configuration Use Count Validation after Removing of Audio Encoder Configuration from Profile.

**Test Case ID:** MEDIA-3-3-8

**ONVIF Core Specification Coverage:** Get media profiles, Remove audio encoder configuration from a profile, Get audio encoder configurations, Get audio encoder configuration

**Command Under Test:** GetProfiles, GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, RemoveAudioEncoderConfiguration

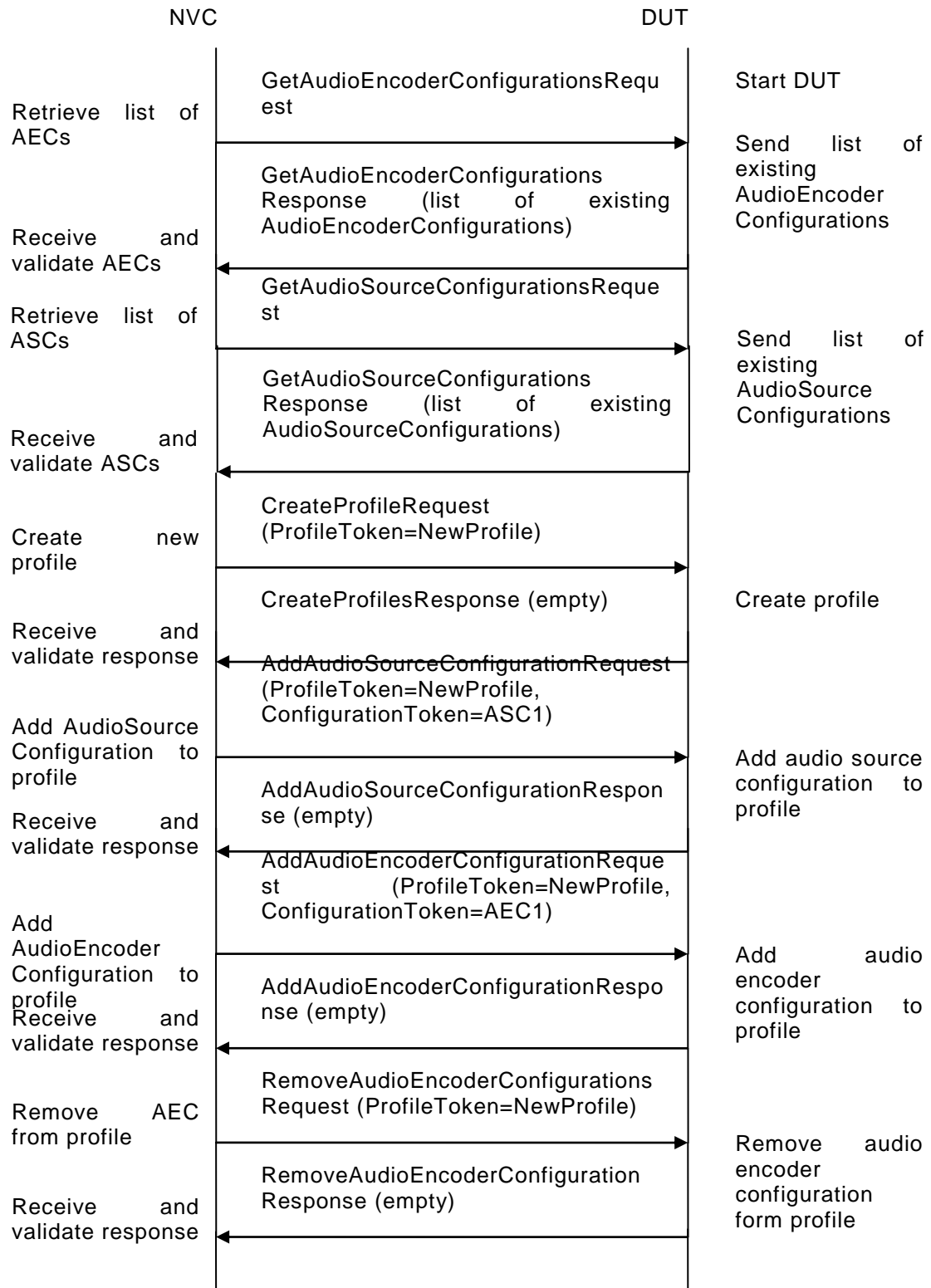
**WSDL Reference:** media.wsdl

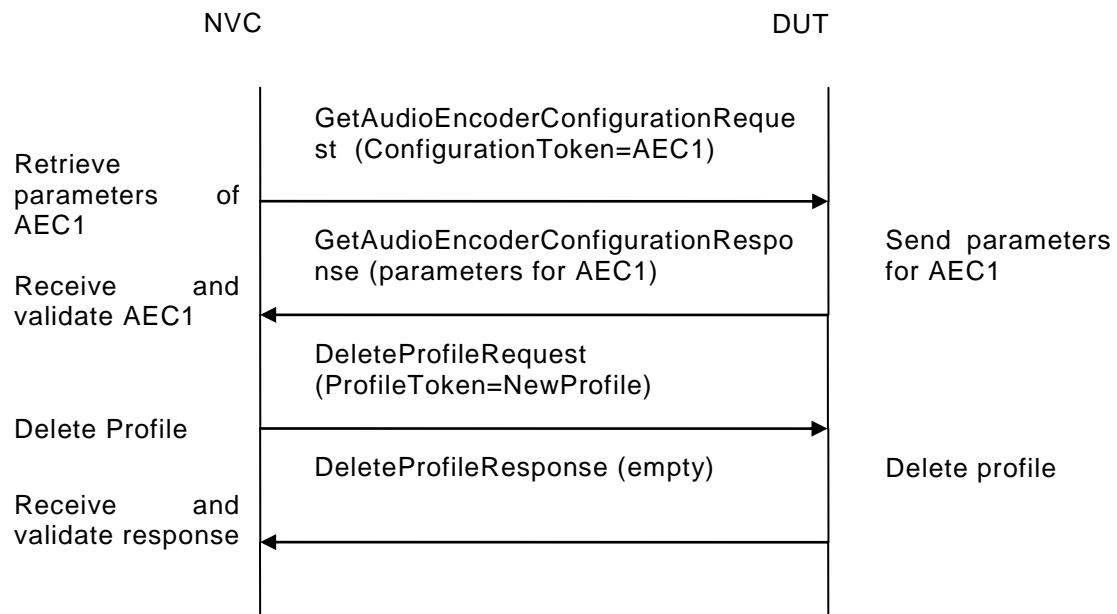
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Encoder Configuration use count after removing audio encoder configuration from profile.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioEncoderConfigurationsRequest` message to retrieve list of audio encoder configurations from device.
4. The DUT sends `GetAudioEncoderConfigurationsResponse` message.
5. NVC invokes `GetAudioSourceConfigurationsRequest` message to retrieve list of audio source configurations from device.
6. The DUT sends `GetAudioSourceConfigurationsResponse` message.
7. NVC invokes `CreateProfileRequest` (**Name=NewName**, **Token=NewProfile**) message to create profile.
8. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
 If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
 NVC invokes `RemoveAudioEncoderConfigurationRequest` (**ProfileToken=profile1**) message to remove `AudioEncoderConfiguration` from profile1  
 The DUT sends `RemoveAudioEncoderConfigurationResponse` message  
 Execute steps 10-18  
 If there is no profile with "fixed" attribute value "false" in profiles list, end test.
9. If the DUT sends `CreateProfileResponse` message, validate `CreateProfileResponse` message from the DUT.

10. NVC invokes AddAudioSourceConfigurationRequest (**ConfigurationToken=first audio source from the list on step 6 (ASC1), ProfileToken=NewProfile**) message to add AudioSourceConfiguration to the new profile.
11. The DUT sends AddAudioSourceConfigurationResponse message.
12. NVC invokes AddAudioEncoderConfigurationRequest (**ConfigurationToken=first audio encoder from the list on step 4 (AEC1), ProfileToken=NewProfile**) message to add AudioEncoderConfiguration to the new profile.
13. The DUT sends AddAudioEncoderConfigurationResponse message.
14. NVC invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken=NewProfile**) message to remove audio encoder configuration from profile1.
15. The DUT sends RemoveAudioEncoderConfigurationResponse message.
16. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
17. The DUT sends GetAudioEncoderConfigurationResponse message.
18. Check that **UseCount=usecount1**, in GetAudioEncoderConfigurationResponse where usecount1 is UseCount value for AEC1 in the list on step 4. If test step 8 is executed, the usecount1 value could be reduced by 1.
19. NVC invokes DeleteProfileRequest (**ProfileToken=NewProfile**) message to remove profile with audio encoder configuration.
20. The DUT sends DeleteProfileResponse message.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send AddAudioEncoderConfigurationResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send RemoveAudioEncoderConfigurationResponse message.

The DUT did not send valid RemoveAudioEncoderConfigurationResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

UseCount value is not decreased by 1 after removing of the AudioEncoderConfiguration from profile.

#### 4.7.9 AUDIO ENCODER CONFIGURATION USE COUNT (DELETION PROFILE WITH AUDIO

## **ENCODER CONFIGURATION)**

**Test Label:** Media Service DUT Audio Encoder Configuration Use Count Validation after Deletion of Profile with Audio Encoder Configuration in it.

**Test Case ID:** MEDIA-3-3-9

**ONVIF Core Specification Coverage:** Get audio encoder configurations, Get audio encoder configuration, Delete media profile

**Command Under Test:** GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, DeleteProfile.

**WSDL Reference:** media.wsdl

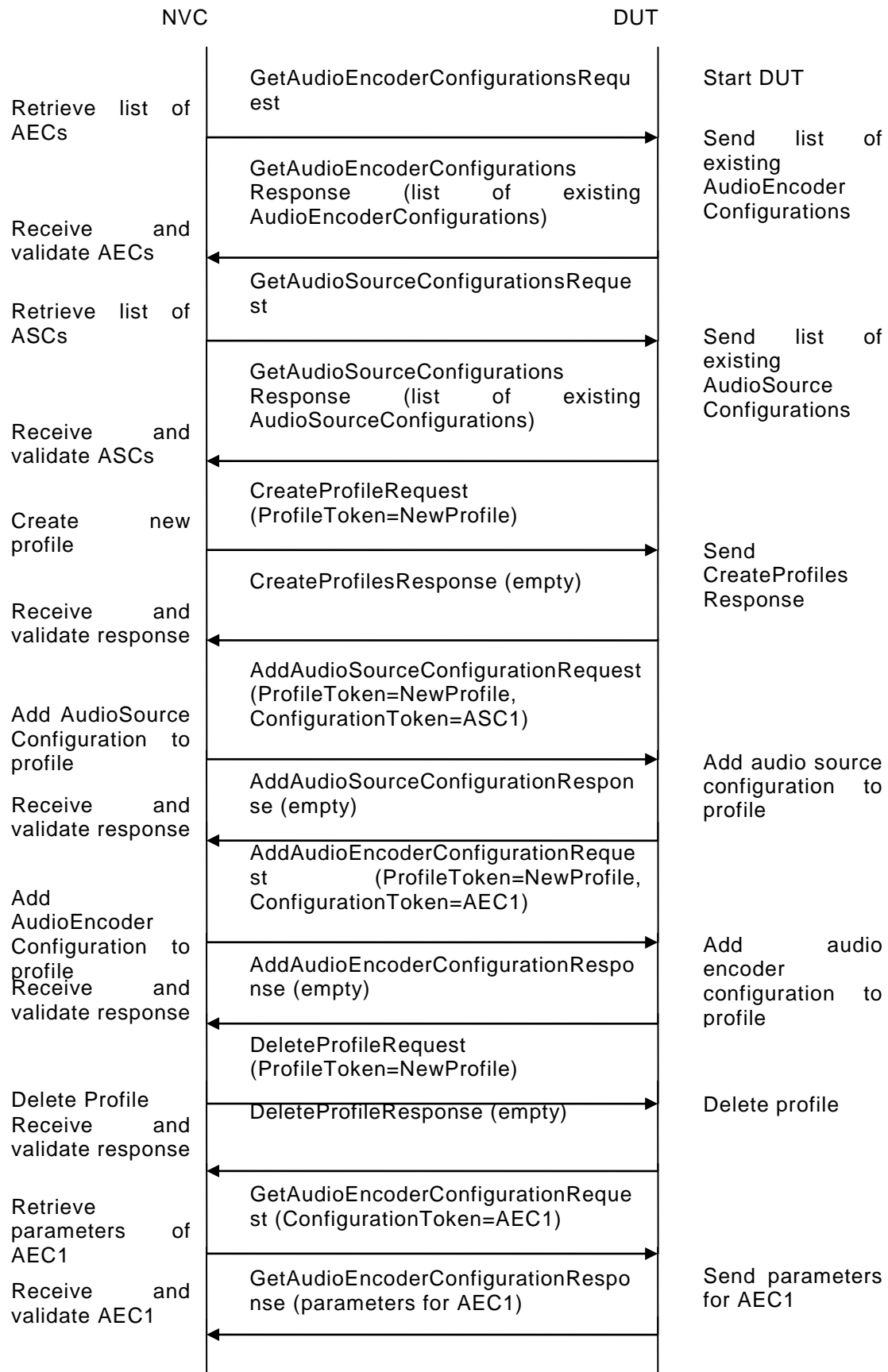
**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Propose:** To check Audio Encoder Configuration use count after deletion profile with audio encoder configuration in it.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes `GetAudioEncoderConfigurationsRequest` message to retrieve list of audio encoder configurations from device.
4. The DUT sends `GetAudioEncoderConfigurationsResponse` message.
5. NVC invokes `GetAudioSourceConfigurationsRequest` message to retrieve list of audio source configurations from device.
6. The DUT sends `GetAudioSourceConfigurationsResponse` message.
7. NVC invokes `CreateProfileRequest` (**Name=NewName, Token=NewProfile**) message to create profile.
8. If the DUT will generate a SOAP 1.2 fault message (**Action/MaxNVTProfiles**),  
 If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)  
 NVC invokes `RemoveAudioEncoderConfigurationRequest` (**ProfileToken=profile1**) message to remove `AudioEncoderConfiguration` from profile1  
 The DUT sends `RemoveAudioEncoderConfigurationResponse` message  
 Execute steps 10-18  
 If there is no profile with "fixed" attribute value "false" in profiles list, end test.
9. If the DUT sends `CreateProfileResponse` message, validate `CreateProfileResponse` message from the DUT.
10. NVC invokes `AddAudioSourceConfigurationRequest` (**ConfigurationToken=first audio source from the list on step 6 (ASC1), ProfileToken=NewProfile**) message to add `AudioConfiguration` to the new profile.
11. The DUT sends `AddAudioSourceConfigurationResponse` message.
12. NVC invokes `AddAudioEncoderConfigurationRequest` (**ConfigurationToken=first audio encoder from the list on step 4 (AEC1), ProfileToken=NewProfile**) message to add `AudioEncoderConfiguration` to the new profile.
13. The DUT sends `AddAudioEncoderConfigurationResponse` message.
14. NVC invokes `DeleteProfileRequest` (**ProfileToken=NewProfile**) message to remove profile with audio encoder configuration.
15. The DUT sends `DeleteProfileResponse` message.
16. NVC invokes `GetAudioEncoderConfigurationRequest` (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
17. The DUT sends `GetAudioEncoderConfigurationResponse` message.
18. Check that **UseCount=usecount1**, in `GetAudioEncoderConfigurationResponse` where usecount1 is UseCount value for AEC1 in the list on step 4. If test step 8 is executed, the usecount1 value could be reduced by 1.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send AddAudioEncoderConfigurationResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

The DUT did not send valid DeleteProfileResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the AudioEncoderConfiguration.

**4.7.10 AUDIO ENCODER CONFIGURATION USE COUNT (SET AUDIO ENCODER CONFIGURATION)**

**Test Label:** Media Audio Encoder Configuration Use Count Validation after Changing of Audio Encoder Configuration.

**Test Case ID:** MEDIA-3-3-10

**ONVIF Core Specification Coverage:** Get media profiles, Get audio encoder configurations, Get audio encoder configuration, Modify audio encoder configurations.

**Command Under Test:** GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, SetAudioEncoderConfiguration

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF SUPPORTED (Audio)

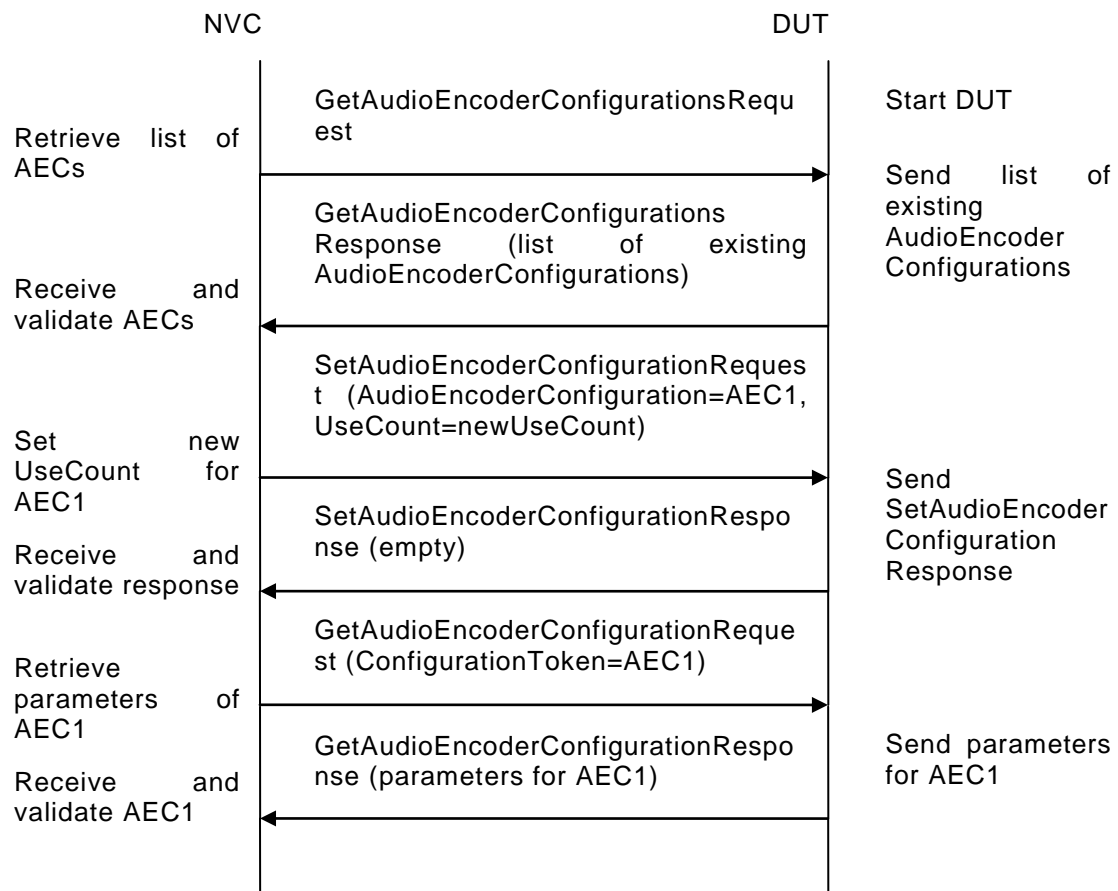
**Test Propose:** To check Audio Encoder Configuration use count after setting audio encoder configuration parameters.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
4. The DUT sends GetAudioEncoderConfigurationsResponse message.
5. NVC invokes SetAudioEncoderConfigurationRequest (**token=the first Audio Encoder Configuration from the list on step 4(AEC1), UseCount=NewUseCount**) message to set parameters for audio encoder configuration.
6. The DUT sends SetAudioEncoderConfigurationResponse message.
7. NVC invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken=AEC1**) message to retrieve audio encoder configuration parameters.
8. The DUT sends GetAudioEncoderConfigurationResponse message.
9. Check that **UseCount =usecount1**, in GetAudioEncoderConfigurationResponse where usecount1 is UseCount value for AEC1 in the list on step 4.

### Test Result:

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send valid SetAudioEncoderConfigurationResponse message.

UseCount value changed after trying to set UseCount value.

## **4.8 PTZ Configuration**

### **4.8.1 PTZ CONFIGURATION**

**Test Label:** Media Configuration PTZ Configuration

**Test Case ID:** MEDIA-4-1-1

**ONVIF Core Specification Coverage:** Get media profiles, Add PTZ configuration to a profile, Remove PTZ configuration from a profile, Delete media profile, GetConfigurations.

**Command Under Test:** None

**WSDL Reference:** media.wsdl

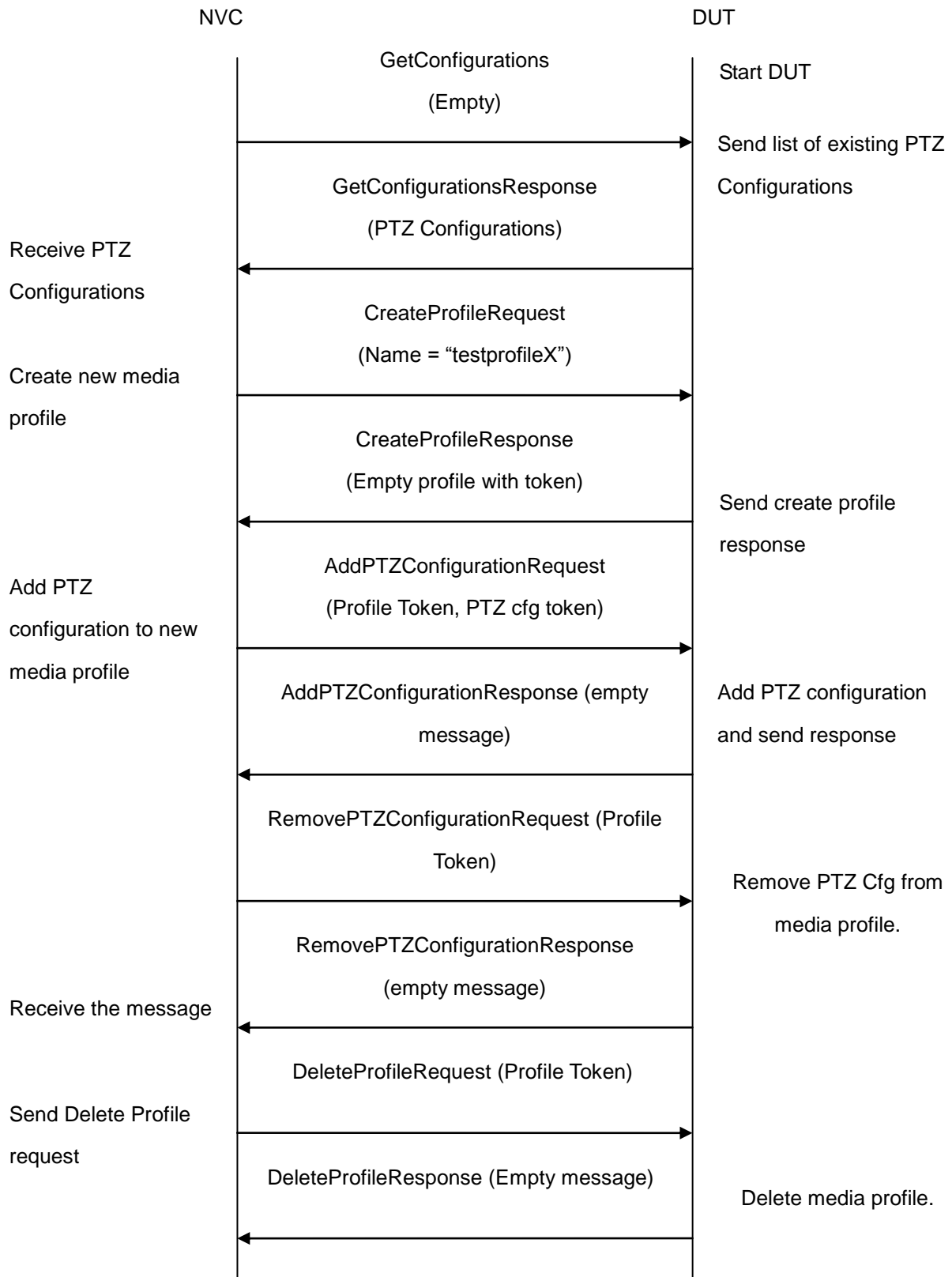
**Requirement Level:** MUST IF SUPPORTED (PTZ)

**Test Purpose:** To verify DUT PTZ Configuration Operations

**Pre-Requisite:** PTZ is supported by DUT

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start the DUT.
3. NVC invokes GetConfigurations request on PTZ service of DUT to retrieve the list of PTZ configurations supported by DUT.
4. NVC validates the GetConfigurationsResponse message sent by the DUT. At least one PTZ configuration should be present in the response.
5. NVC invokes CreateProfile request with **ProfileToken** = 'testprofileX'.
6. DUT creates new media profile and sends the response.
7. NVC invokes AddPTZConfiguration request with **ProfileToken** as 'testprofileX' and **ConfigurationToken** as one of the existing PTZConfiguration tokens.
8. DUT adds the PTZ configuration to the profile and sends the response.
9. NVC invokes RemovePTZConfiguration request with **ProfileToken** as 'testprofileX'.
10. DUT removes the PTZ configuration from media profile and sends the response.
11. NVC invokes DeleteProfile request with **ProfileToken** as 'testprofileX'.
12. DUT deletes the media profile and sends the response.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send valid GetConfigurationsResponse message.

DUT doesn't have PTZ configuration.

DUT did not send AddPTZConfigurationResponse message.

DUT did not send RemovePTZConfigurationResponse message.

DUT did not send DeleteProfileResponse message.

**4.8.2 PTZ CONFIGURATIONS AND PROFILES CONSISTENCY**

**Test Label:** Media Service and PTZ Service DUT GetConfigurations Command and GetProfiles Command Consistency Validation.

**Test Case ID:** MEDIA-4-1-2

**ONVIF Core Specification Coverage:** Get media profiles, GetConfigurations.

**Command Under Test:** GetProfiles, GetConfigurations

**WSDL Reference:** media.wsdl, ptz.wsdl.

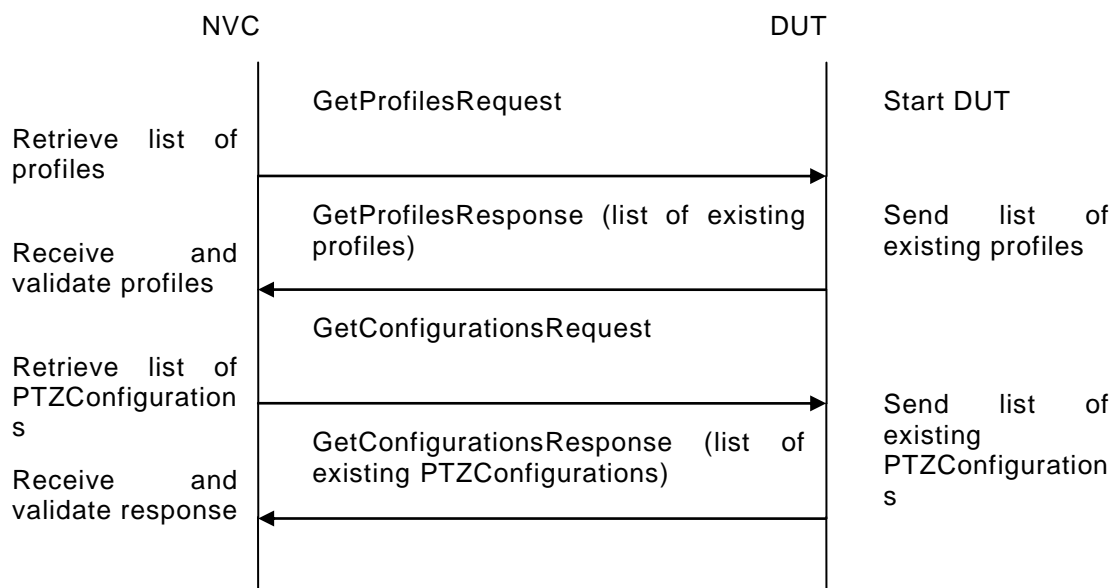
**Requirement Level:** MUST IF SUPPORTED (PTZ)

**Test Propose:** To check that GetPTZConfigurations command and GetProfiles command are consistent.

**Pre-Requisite:** PTZ is supported by DUT. NVC gets the Media Service entry point and PTZ Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC invokes GetProfilesRequest message to retrieve list of profiles and their PTZ configurations.
4. The DUT sends GetProfilesResponse message.
5. NVC invokes GetConfigurations message to retrieve list of PTZ Configurations from device.
6. The DUT sends GetConfigurationsResponse message.
7. Check that each PTZConfiguration from GetConfigurationsResponse message has unique token.
8. Check that all PTZConfigurations from the GetProfilesResponse message are included in list the GetConfigurationsResponse message.
9. Check that PTZConfiguration parameters are same in the GetProfilesResponse message and in the GetConfigurationsResponse message.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetConfigurationsResponse message.

The DUT did not send valid GetConfigurationsResponse message.

The DUT return two or more PTZConfigurations in GetConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with PTZConfigurations that were not included in the GetConfigurationsResponse message.

The DUT returned different parameters list and parameters values for the same PTZConfiguration in the GetConfigurationsResponse message and in the GetProfilesResponse message.

## **4.9 Metadata Configuration**

### **4.9.1 METADATA CONFIGURATION**

**Test Label:** Media Configuration Metadata Configuration

**Test Case ID:** MEDIA-5-1-1

**ONVIF Core Specification Coverage:** Create media profile, Add metadata configuration to a profile, Remove metadata configuration from a profile, Delete media profile, Get metadata configurations, Get metadata configuration, Get compatible metadata configurations, Get metadata configuration options, Modify a metadata configuration.

**Command Under Test:** None

**WSDL Reference:** media.wsdl

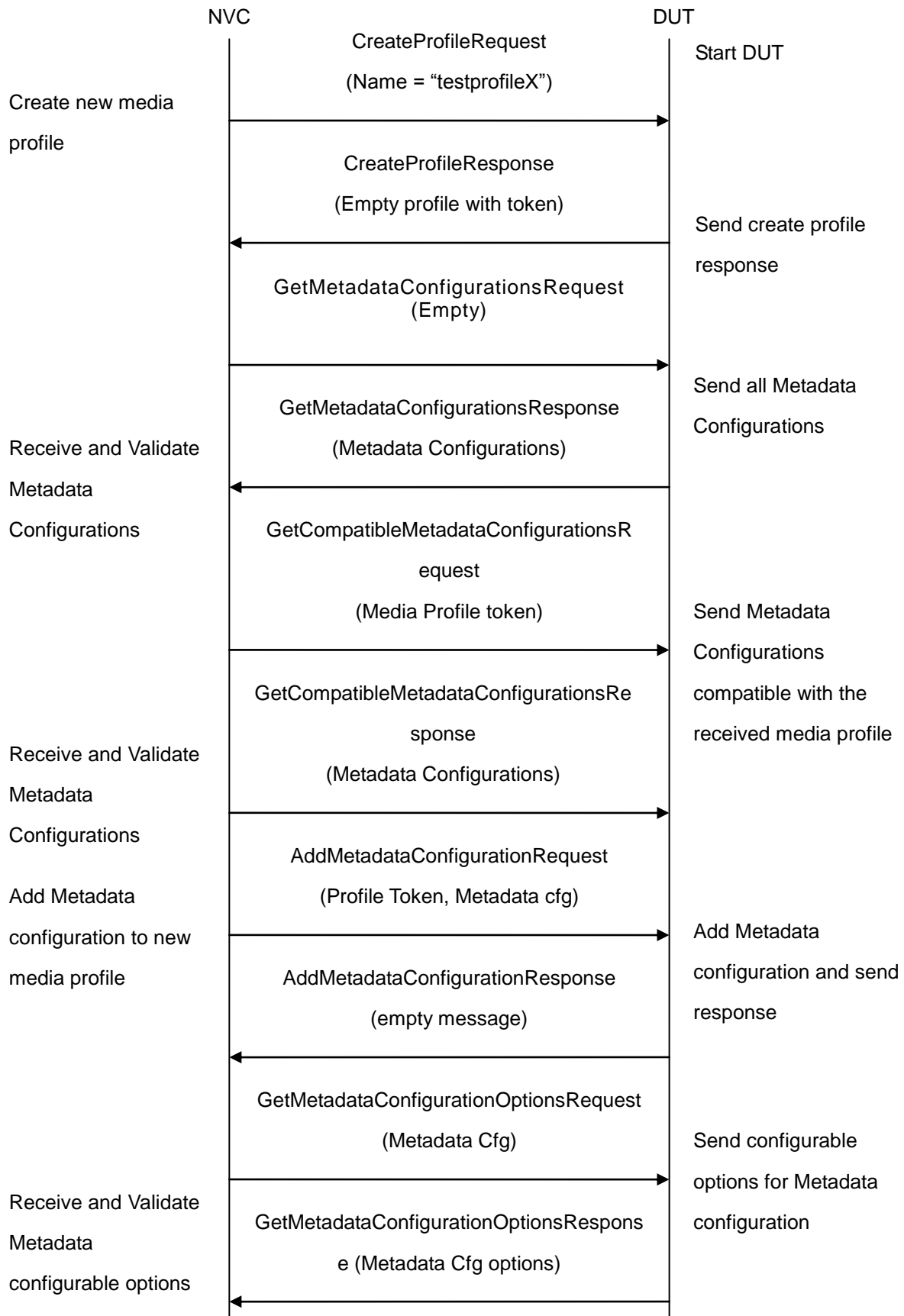
**Requirement Level:** MUST

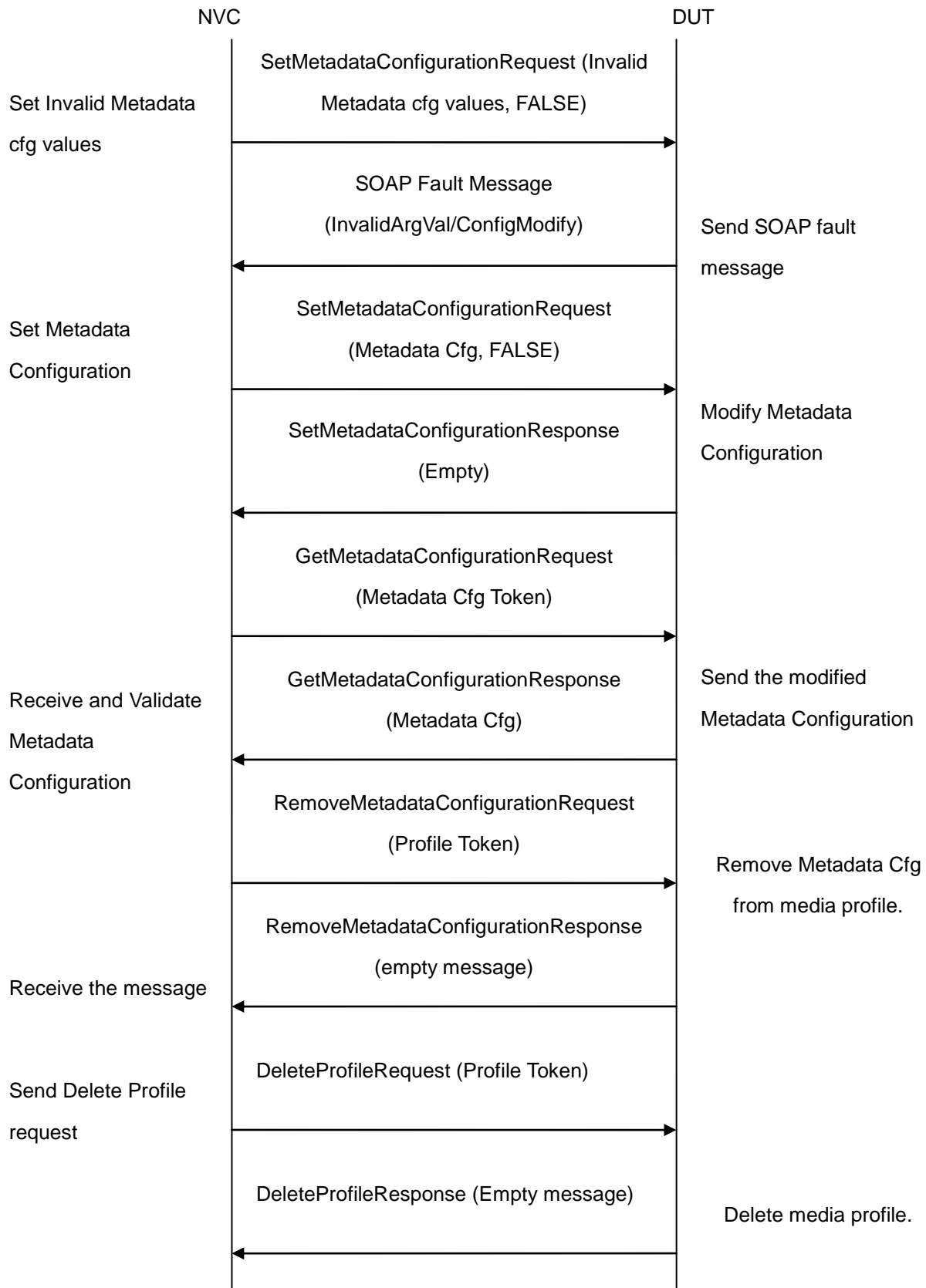
**Test Purpose:** To verify DUT Metadata Configuration Operations

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**







**Test Procedure:**

1. Start an NVC.
2. Start the DUT.
3. NVC invokes CreateProfile request with **ProfileToken** = '**testprofileX**'.
4. DUT creates new media profile and sends the response.
5. NVC will invoke GetMetadataConfigurations request to retrieve the list of metadata configurations supported by the DUT.
6. NVC verifies the list of metadata configurations sent by the DUT.
7. NVC invokes GetCompatibleMetadataConfigurations request with '**testprofileX**' as **ProfileToken**.
8. DUT sends the list of metadata configurations compatible with the received media profile token.
9. NVC invokes AddMetadataConfiguration request message with **ProfileToken** as '**testprofileX**' and **ConfigurationToken** as one of the metadata configuration tokens received in the GetCompatibleMetadataConfigurations response.
10. DUT adds the Metadata configuration to the profile and sends the response.
11. NVC invokes GetMetadataConfigurationOptions request with **ConfigurationToken** as same token in AddMetadataConfiguration request.
12. DUT sends the configurable options supported for the received metadata configuration.
13. NVC invokes SetMetadataConfiguration request with metadata configuration values outside the range defined in GetMetadataConfigurationOptions response and '**ForcePersistence**' flag as '**FALSE**'.
14. DUT send the SOAP 1.2 fault message (**InvalidArgVal/ConfigModify**).
15. NVC verifies the SOAP fault message sent by DUT.
16. NVC invokes SetMetadataConfiguration request with metadata configuration values within the range defined in GetMetadataConfigurationOptions response and '**ForcePersistence**' flag as '**FALSE**'.
17. DUT modifies the metadata configuration and sends the SetMetadataConfiguration indicating success.
18. NVC verifies the modified Metadata configuration by invoking the GetMetadataConfiguration request.
19. DUT sends the modified Metadata configuration in GetMetadataConfiguration response.
20. NVC invokes RemoveMetadataConfiguration request with **ProfileToken** as '**testprofileX**'.
21. DUT removes the Metadata configuration token from media profile and sends the response.
22. NVC invokes DeleteProfile request with **ProfileToken** as '**testprofileX**'.
23. DUT deletes the media profile and sends the response.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send CreateProfileResponse message.

DUT did not send GetMetadataConfigurationsResponse message.

DUT did not send GetCompatibleMetadataConfigurationsResponse message.

DUT did not send AddMetadataConfigurationResponse message.

DUT did not send GetMetadataConfigurationOptionsResponse message.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetMetadataConfiguration request.

DUT did not send SetMetadataConfigurationResponse message.

DUT did not send GetMetadataConfigurationResponse message.

DUT did not modify metadata configuration correctly.

DUT did not send RemoveMetadataConfigurationResponse message.

DUT did not send DeleteProfileResponse message.

## **4.10 Media Streaming**

### **4.10.1 SNAPSHOT URI**

**Test Label:** Media Configuration Snapshot URI

**Test Case ID:** MEDIA-6-1-1

**ONVIF Core Specification Coverage:** Get media profiles, Request snapshot URI.

**Command Under Test:** GetSnapshotUri

**WSDL Reference:** media.wsdl

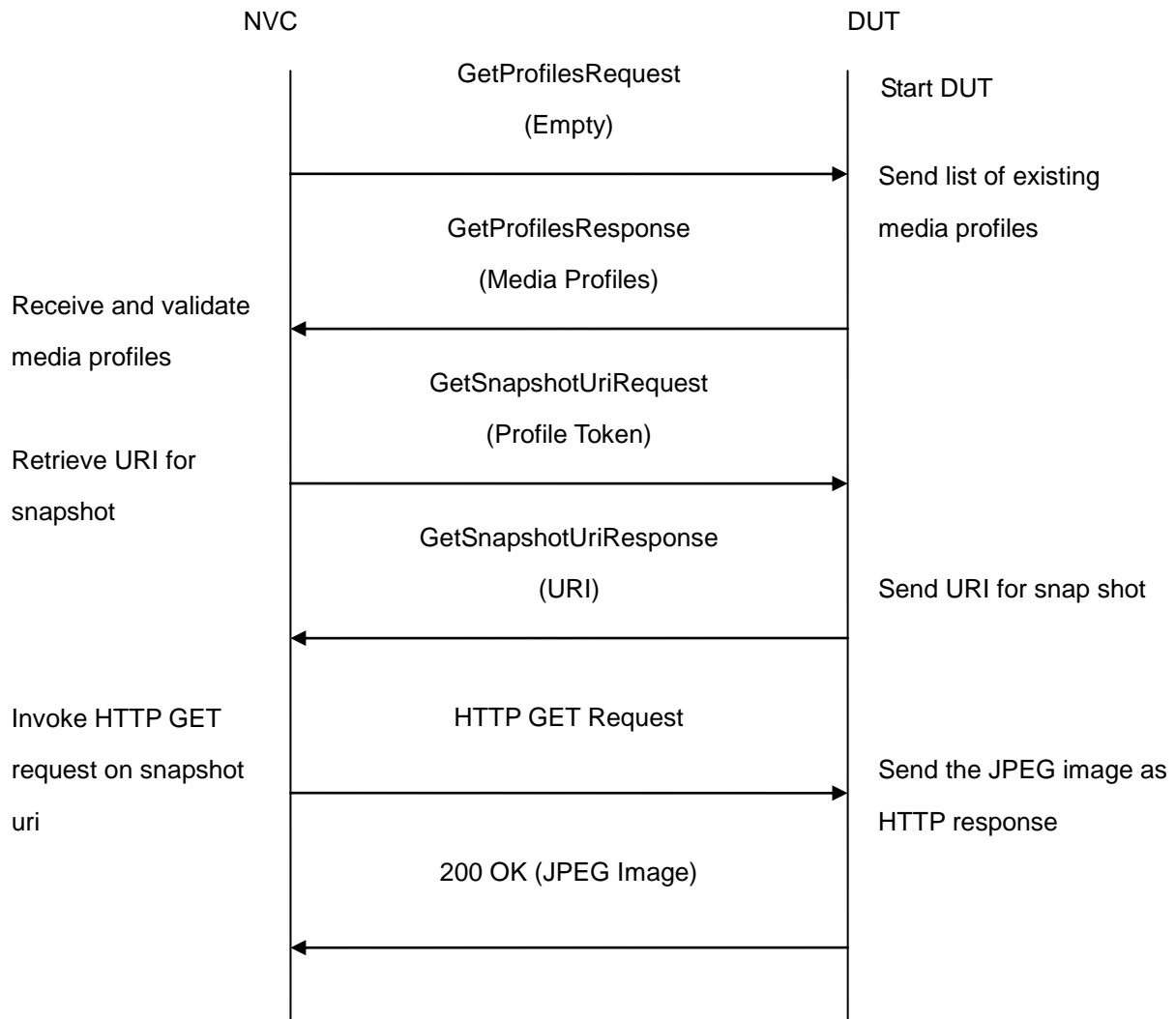
**Requirement Level:** SHOULD

**Test Purpose:** To retrieve snapshot URI of DUT for given media profile

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC invokes GetProfiles request to retrieve the list of existing media profiles on DUT.
4. NVC validates the GetProfilesResponse message sent by the DUT. At least one media profile should be present with video source and video encoder.
5. NVC invokes GetSnapshotUri request with **ProfileToken** as one of the media profile tokens received in GetProfilesResponse message.
6. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetSnapshotUriResponse message.
7. NVC invokes HTTP GET request on the snapshot URI sent by DUT.
8. DUT sends 200 OK message and the single shot JPEG image data.

9. NVC verifies the JPEG image sent by the DUT.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send GetProfilesResponse message.

DUT did not send GetSnapshotUriResponse message.

DUT did not send one or more mandatory parameters in the GetSnapshotUriResponse message (mandatory parameters –Uri, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send 200 OK message.

DUT did not send valid JPEG image data.

## **4.11 Error Handling**

### **4.11.1 SOAP FAULT MESSAGE**

**Test Label:** Media Configuration DUT generates SOAP 1.2 fault message fo Invalid GetStreamUriRequest Message.

**Test Case ID:** MEDIA-7-1-1

**ONVIF Core Specification Coverage:** Request stream URI.

**Command Under Test:** GetStreamUri

**WSDL Reference:** media.wsdl

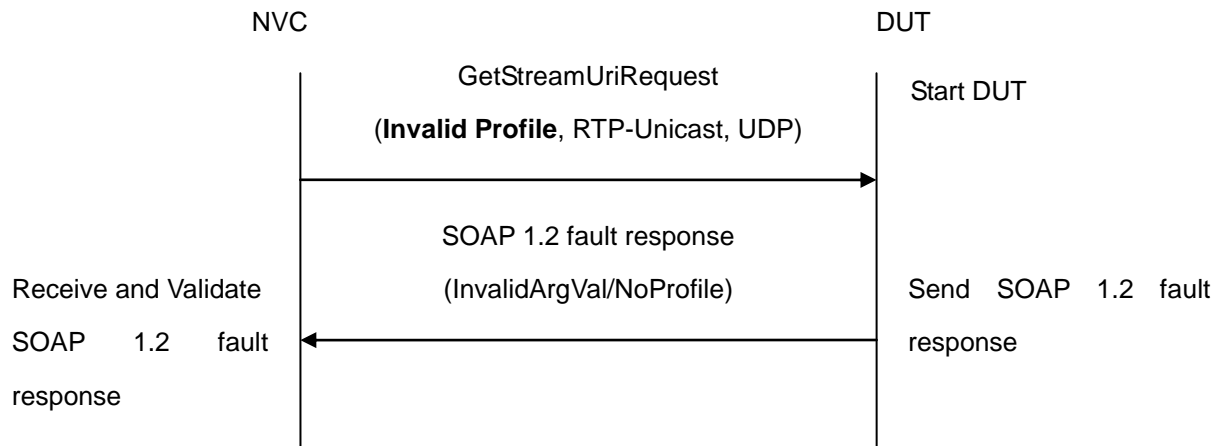
**Requirement Level:** SHOULD

**Test Purpose:** To verify that DUT generates SOAP 1.2 fault message to the invalid GetStreamUriRequest message (Invalid Media Profile).

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**



#### Test Procedure:

1. Start an NVC.
2. Start the DUT.
3. NVC will invoke GetStreamUriRequest message with **invalid media profile**.
4. DUT will generate the SOAP 1.2 fault message (**InvalidArgVal/NoProfile**).

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct SOAP 1.2 fault message (fault code, namespace etc).

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

#### 4.11.2 SOAP FAULT MESSAGE

**Test Label:** Media Configuration DUT generates SOAP 1.2 fault message for Invalid GetStreamUriRequest Message.

**Test Case ID:** MEDIA-7-1-2

**ONVIF Core Specification Coverage:** Request stream URI.

**Command Under Test:** GetStreamUri

**WSDL Reference:** media.wsdl

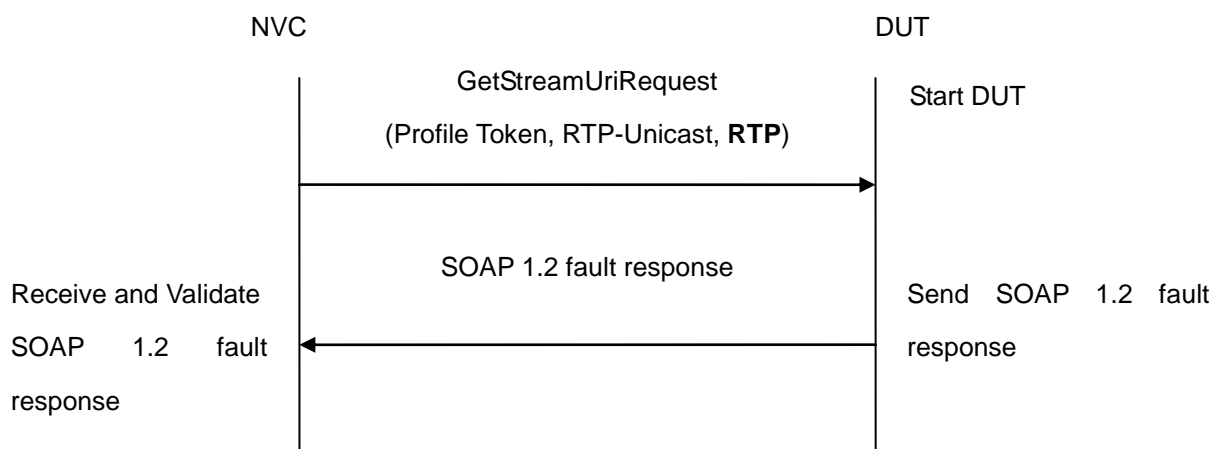
**Requirement Level:** SHOULD

**Test Purpose:** To verify that DUT generates SOAP 1.2 fault message to the invalid GetStreamUriRequest message (Invalid Transport).

**Pre-Requisite:** None

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start an DUT.
3. NVC will invoke GetStreamUriRequest message with invalid Transport (**RTP**).
4. DUT will generate the SOAP 1.2 fault message.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct SOAP 1.2 fault message (fault code, namespace etc).

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

#### 4.11.3 START MULTICAST - INVALID PROFILE TOKEN

**Test Label:** Media DUT StartMulticastStreaming Command Validation (Invalid **ProfileToken**)

**Test Case ID:** MEDIA-7-1-3

**ONVIF Core Specification Coverage:** Start multicast streaming

**Command Under Test:** StartMulticastStreaming

**WSDL Reference:** media.wsdl

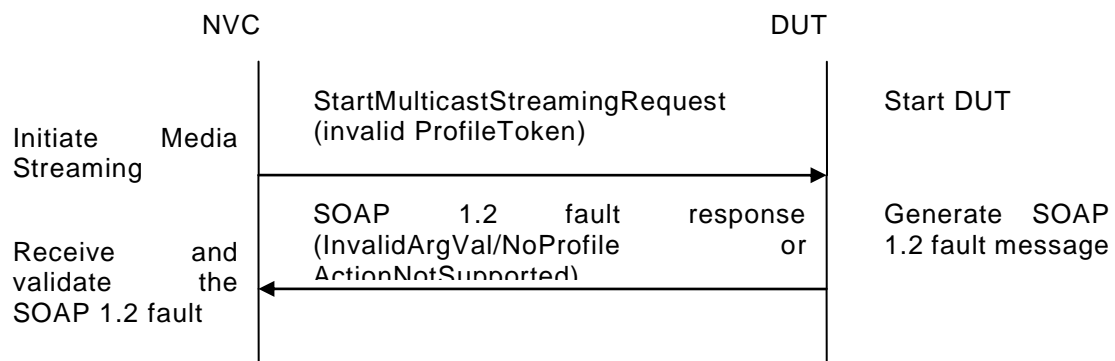
**Requirement Level:** MUST

**Test Propose:** To verify the behavior of StartMulticastStreaming command in case of invalid Profile Token.

**Pre-Requisite:** NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an NVC.
2. Start a DUT.
3. NVC will invoke StartMulticastStreamingRequest message (**invalid ProfileToken**).
4. The DUT will generate a SOAP 1.2 fault message (**InvalidArgVal/NoProfile** or **ActionNotSupported** (for the case when multicast is not supported)).

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

**Note:** See Annex in [ONVIF Base Test] for Invalid SOAP 1.2 fault message definition.

**Note:** Other faults than specified in the test are acceptable but specified are preferable.

## **5 Real Time Streaming Test Cases**

### **5.1 Video Streaming**

#### **5.1.1 MEDIA CONTROL – RTSP/TCP**

**Test Label:** Real Time Viewing DUT RTSP control messages.

**Test Case ID:** RTSS-1-1-1

**ONVIF Core Specification Coverage:** Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST

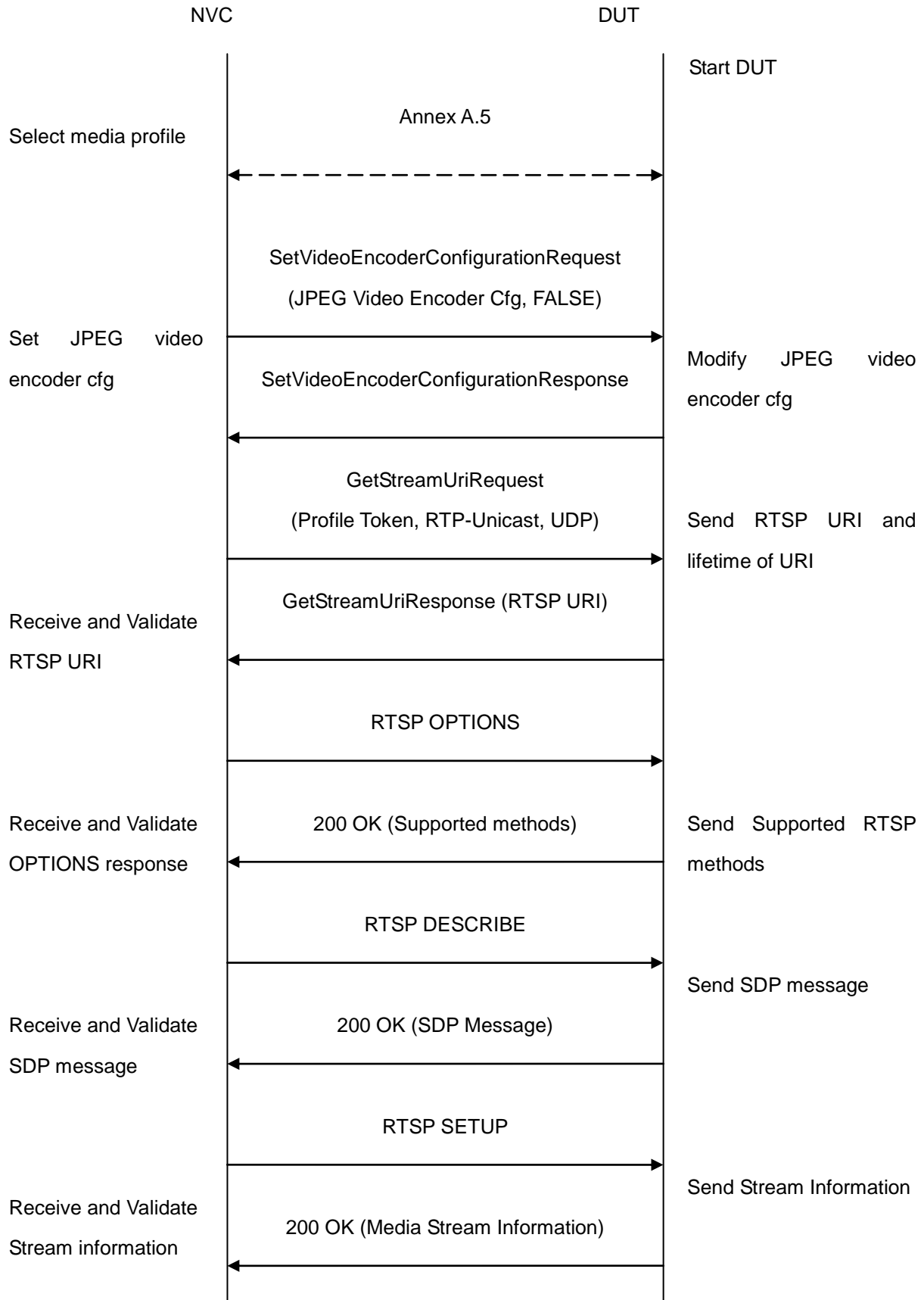
**Test Purpose:** To verify RTSP control messages of DUT.

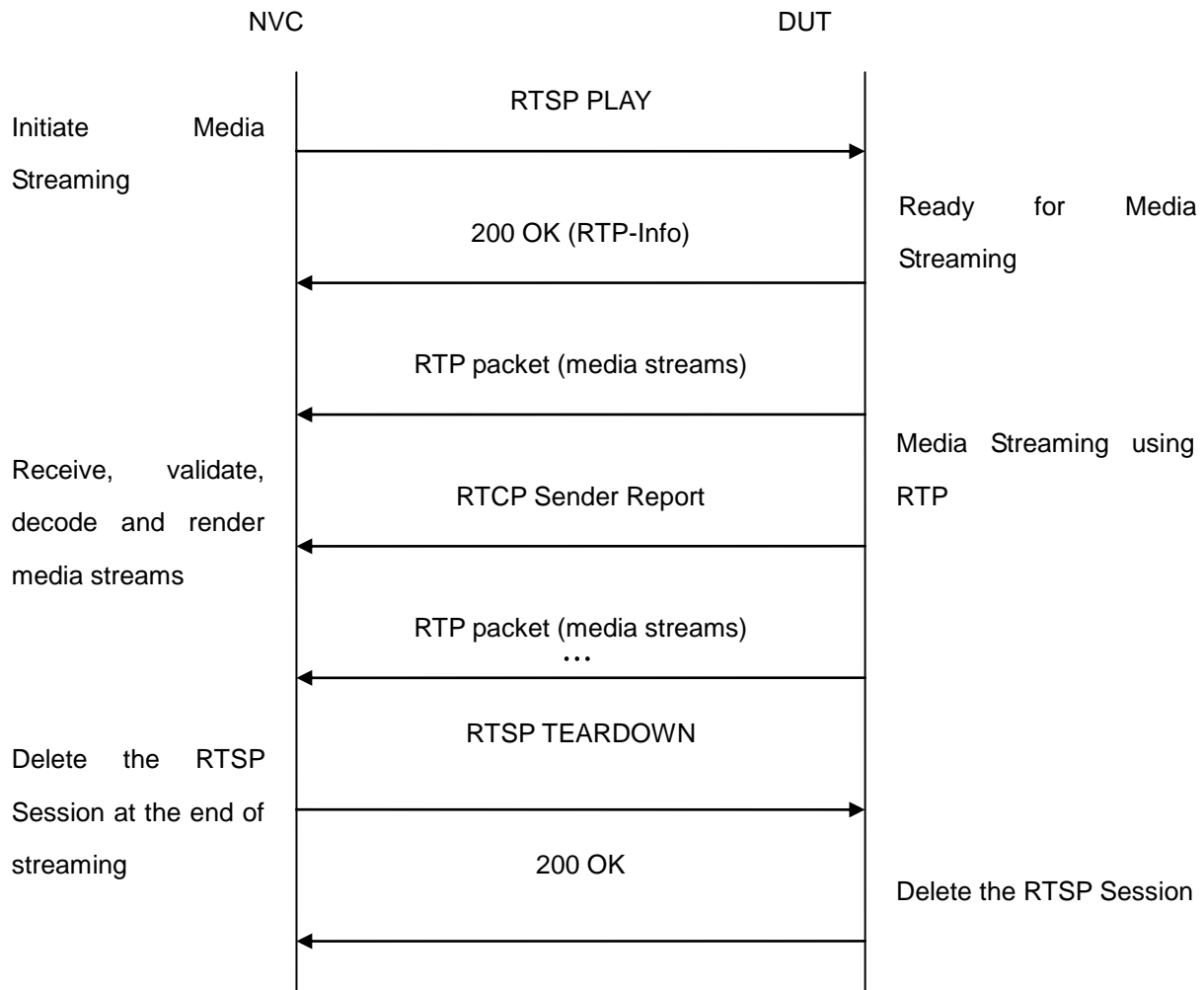
**Pre-Requisite:** A media profile with JPEG video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to **retrieve** media stream URI for the selected media profile.

7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC will invoke RTSP OPTIONS control request to understand the RTSP methods supported by DUT.
10. DUT sends 200 OK Response and list of supported RTSP methods.
11. NVC will invoke RTSP DESCRIBE control request to retrieve the media description information.
12. DUT sends 200 OK Response and SDP message.
13. NVC validates the session description information in the SDP message.
14. NVC will invoke RTSP SETUP control request to create a RTSP Session.
15. DUT sends 200 OK Response and Stream Information details.
16. NVC Verifies "Transport", "Session" and "timeout" header fields in the SETUP response message.
17. NVC will invoke RTSP PLAY control request to initiate the media streaming.
18. DUT sends 200 OK Response and RTP protocol information.
19. NVC verifies "Session", "RTP-Info", "seq", "uri" and "rtptime" header fields in the PLAY response message.
20. DUT transfers media streams over RTP/UDP.
21. DUT sends RTCP sender report to NVC.
22. NVC validates RTCP packets.
23. NVC validates RTP header for each media stream and render it after the validation.
24. NVC will invoke RTSP TEARDOWN control request to terminate the RTSP session at the end of the streaming.
25. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send mandatory RTSP commands DESCRIBE, SETUP, PLAY, TEARDOWN and SET\_PARAMETER in OPTIONS response.

DUT did not send correct media stream information in the SDP message.

DUT did not send mandatory headers or fields in the SETUP response message.

DUT did not send mandatory headers or fields in the PLAY response message.

DUT did not send RTSP 200 OK response for RTSP OPTIONS, DESCRIBE, SETUP and PLAY requests.

RTSP Session is terminated by DUT during media streaming.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.1.2 MEDIA STREAMING – RTSP KEEPALIVE (SET\_PARAMETER)

**Test Label:** Real Time Viewing DUT RTSP Keep-alive.

**Test Case ID:** RTSS-1-1-2

**ONVIF Core Specification Coverage:** Keep-alive method for RTSP session.

**Command Under Test:** None

**WSDL Reference:** None

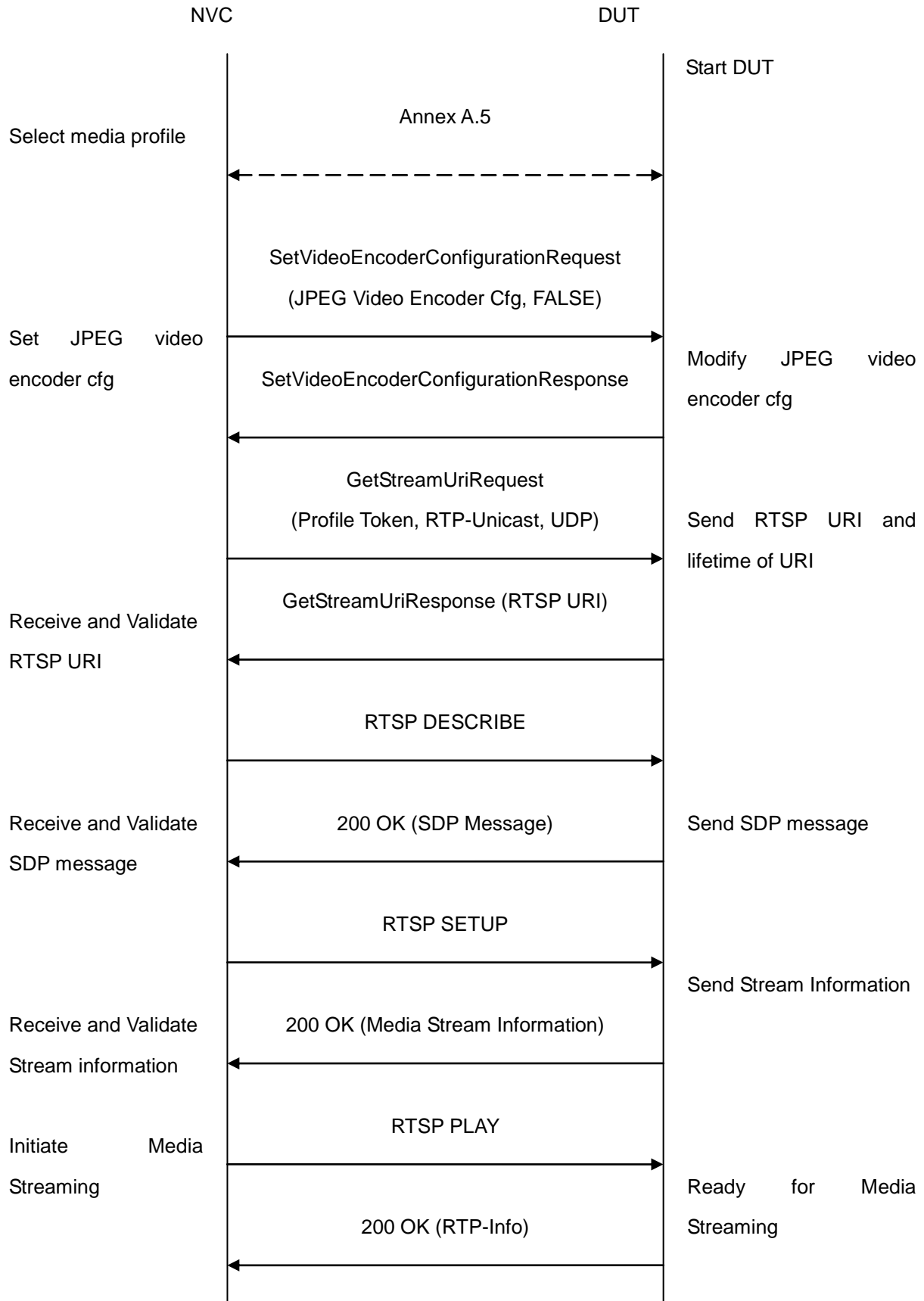
**Requirement Level:** MUST

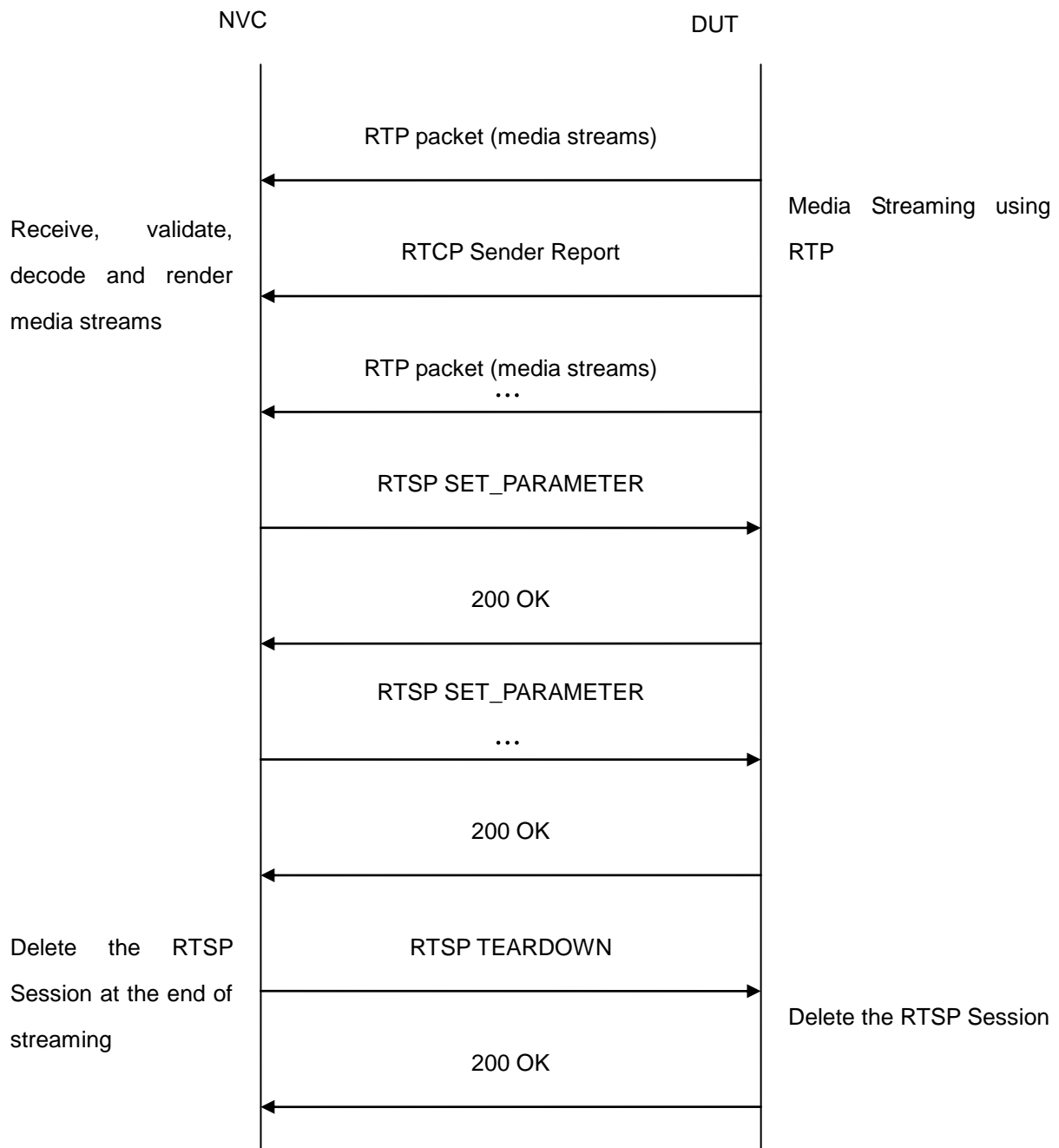
**Test Purpose:** To verify NVC and DUT exchange SET\_PARAMETER messages during an active streaming session.

**Pre-Requisite:** A media profile with JPEG video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start an DUT.

3. NVC selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC will invoke RTSP control requests (**DESCRIBE, SETUP and PLAY**).
10. NVC will verify "**Timeout**" header in the SETUP Response from DUT.
11. Based on the "**Timeout**" value, NVC will invoke **RTSP SET\_PARAMETER** messages.
12. DUT will respond with 200 OK for RTSP SET\_PARAMETER request.
13. Verify that the NVC and DUT are exchanging periodic SET\_PARAMETER messages while a stream is being delivered.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send Timeout header in RTSP SETUP RESPONSE.

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and SET\_PARAMETER requests.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

### 5.1.3 MEDIA STREAMING - RTSP KEEPALIVE (OPTIONS)

**Test Label:** Real Time Viewing DUT RTSP Keep-alive with Options Command.

**Test Case ID:** RTSS-1-1-3

**ONVIF Core Specification Coverage:** Keep-alive method for RTSP session

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** SHALL

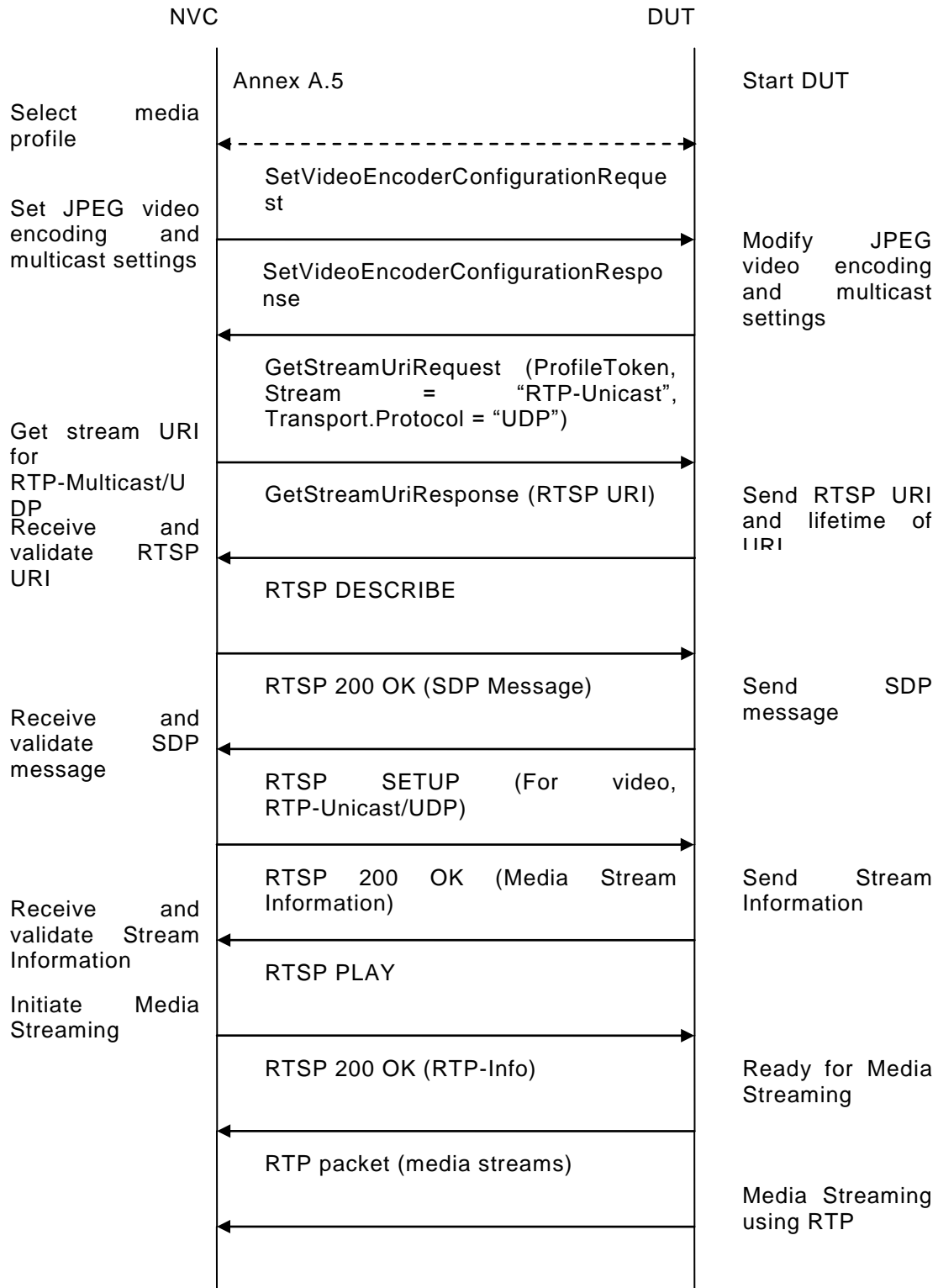
**Test Propose:** To verify NVC and DUT exchange OPTIONS messages during an active streaming session.

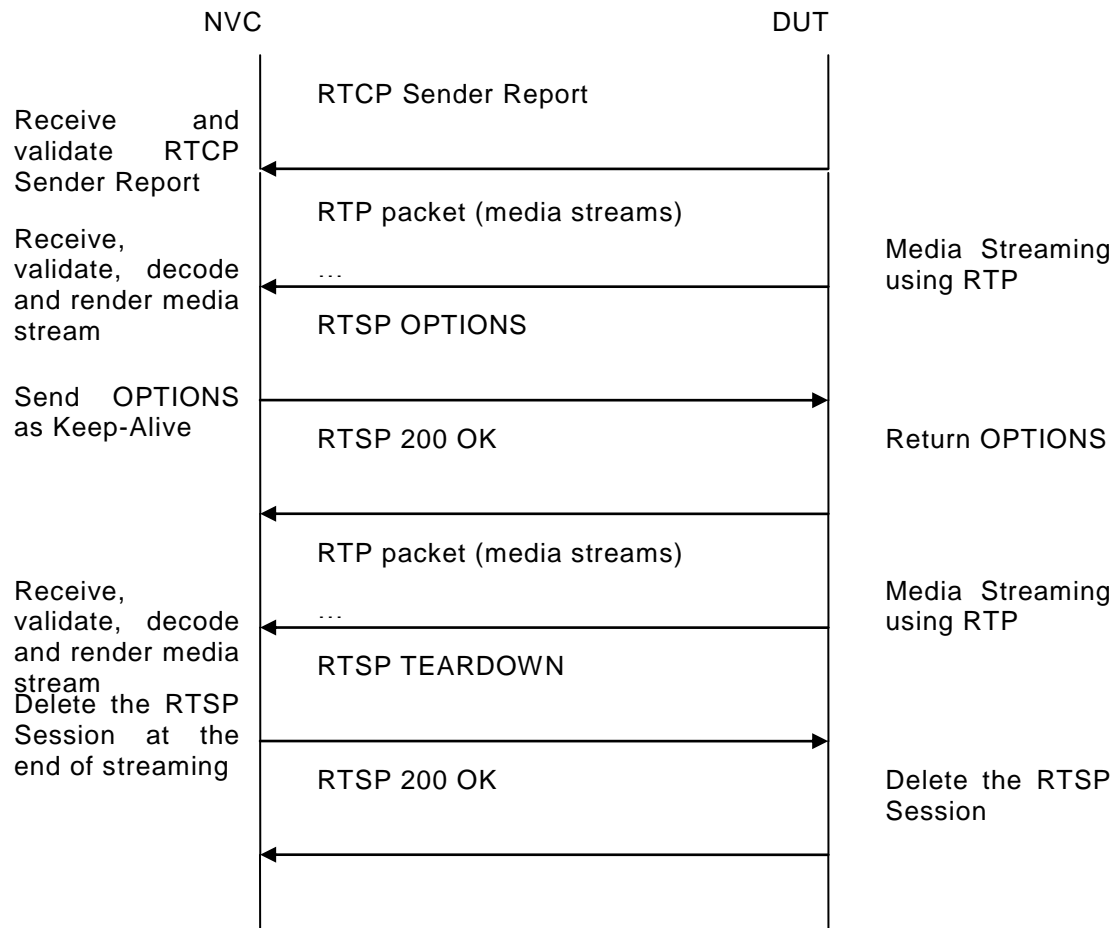
**Test Configuration:** NVC and DUT

**Pre-Requisite:** A media profile with JPEG video encoder configuration.

**Test Sequence:**







### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. Verify the SetVideoEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.
7. Verify the GetStreamUriResponse message from the DUT.
8. Verify the RTSP media stream URI provided by the DUT.
9. NVC will invoke RTSP control requests (DESCRIBE, SETUP and PLAY).

10. Verify "Timeout" header in the SETUP Response from DUT.
11. Based on the "Timeout" value, NVC will invoke RTSP OPTIONS messages.
12. DUT will respond with 200 OK for RTSP OPTIONS request.
13. Verify that the NVC and DUT are exchanging periodic OPTIONS messages while a stream is being delivered.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send Timeout header in RTSP SETUP RESPONSE.

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and OPTIONS requests.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

#### 5.1.4 MEDIA STREAMING – JPEG (RTP-Unicast / UDP)

**Test Label:** Real Time Viewing DUT JPEG media streaming using RTP-Unicast/UDP transport.

**Test Case ID:** RTSS-1-1-4

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

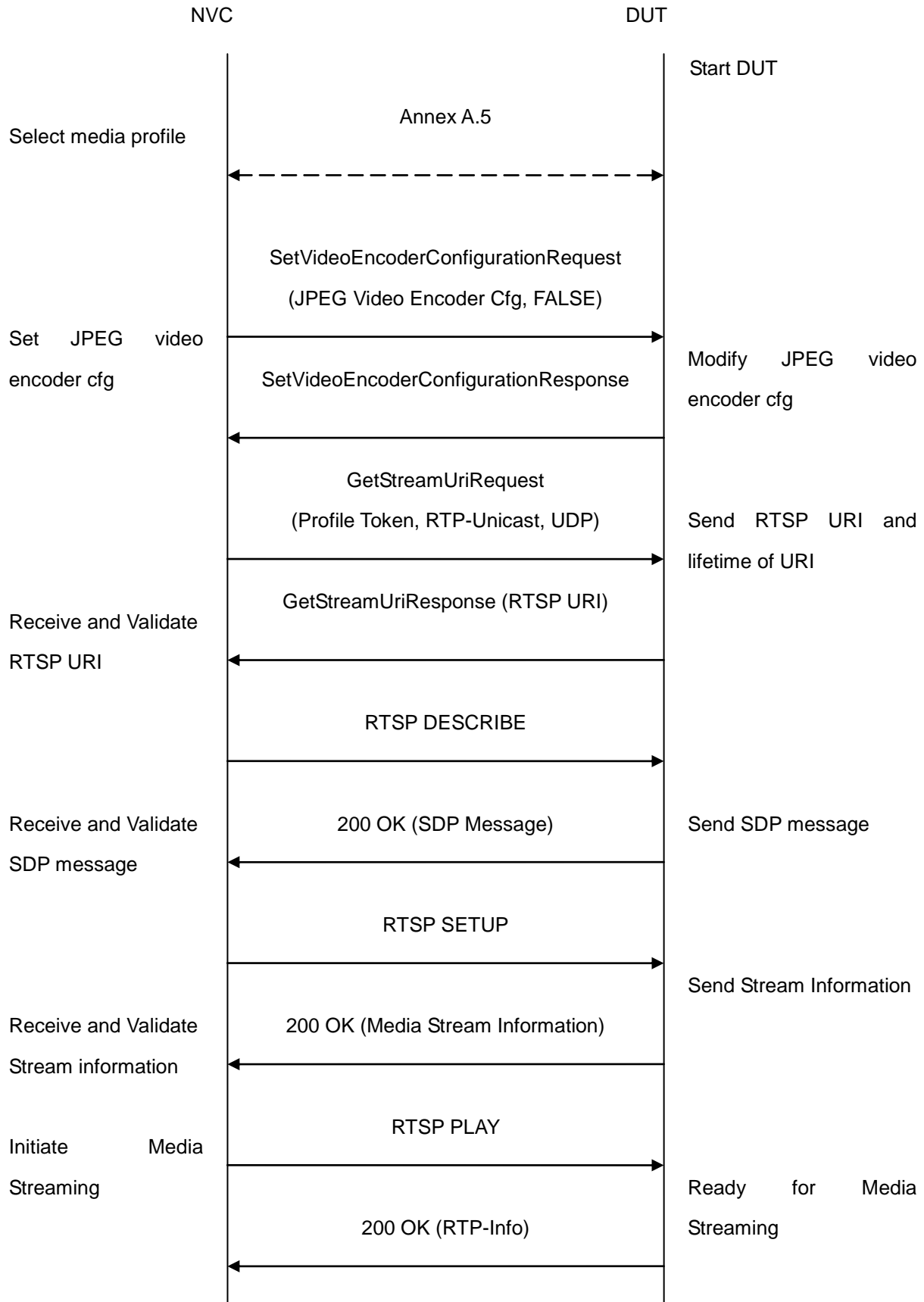
**Requirement Level:** MUST

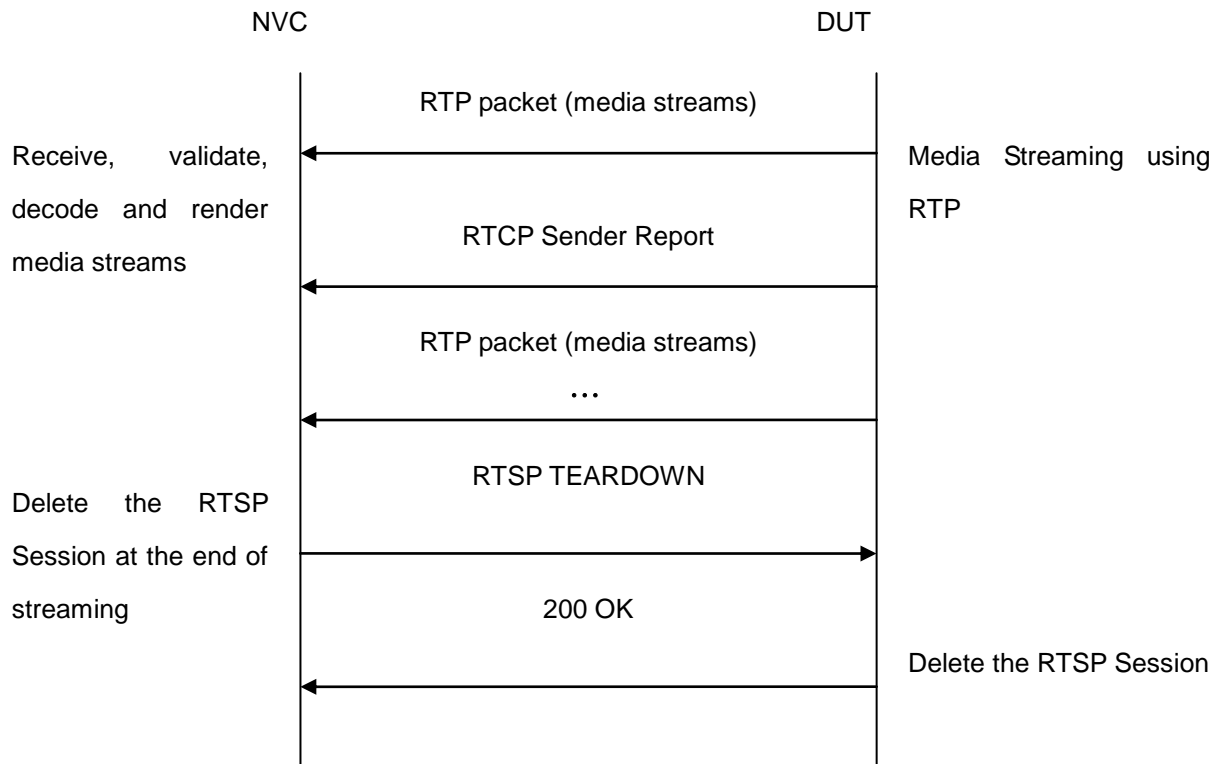
**Test Purpose:** To verify JPEG media streaming based on RTP/UDP Unicast Transport.

**Pre-Requisite:** A media profile with JPEG video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP**.

12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends JPEG RTP media stream to NVC over UDP.
16. DUT sends RTCP sender report to NVC.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

**5.1.5 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT JPEG media streaming using HTTP transport.

**Test Case ID:** RTSS-1-1-5

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST

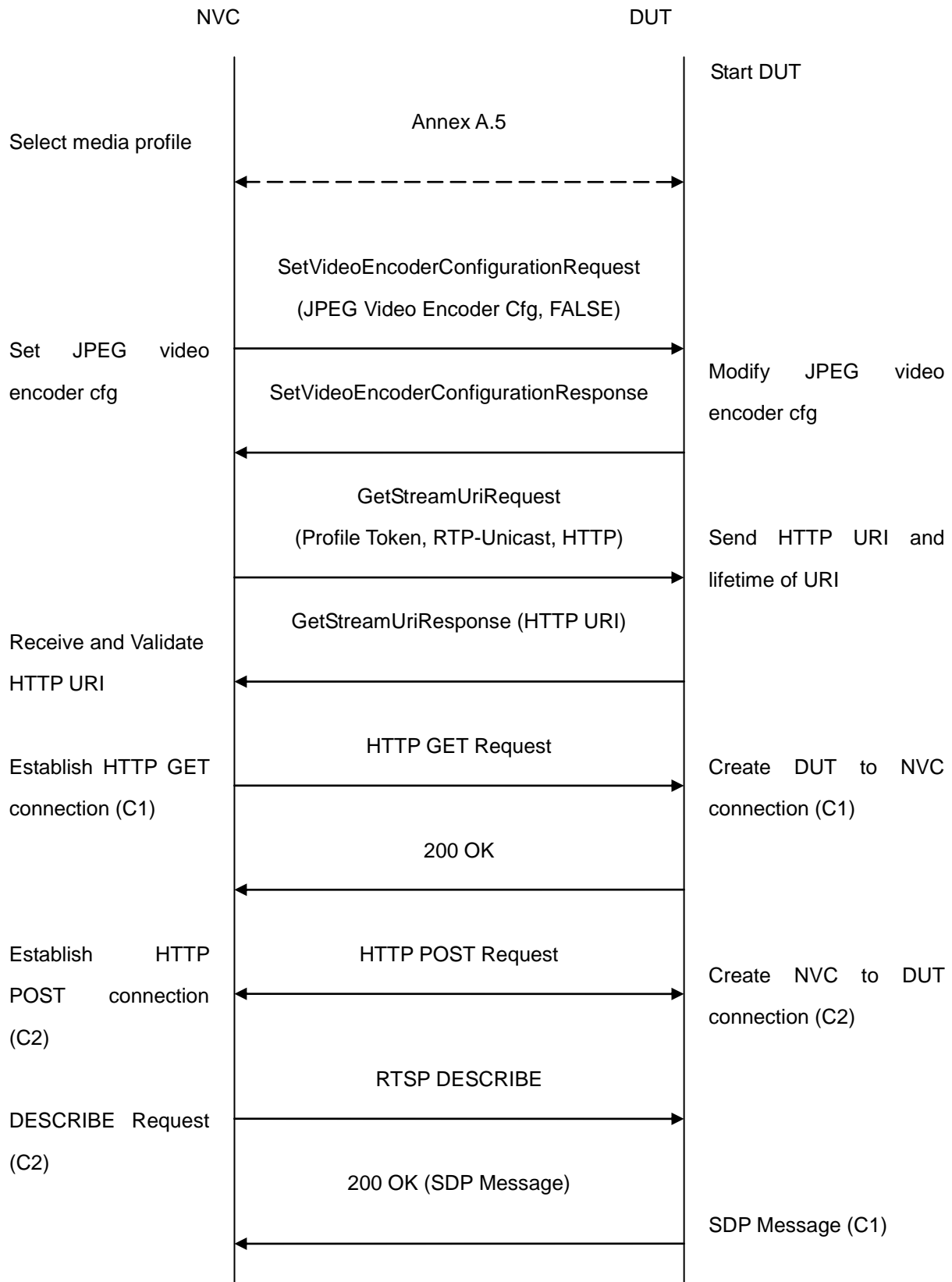
**Test Purpose:** To verify JPEG media streaming based on HTTP Transport.

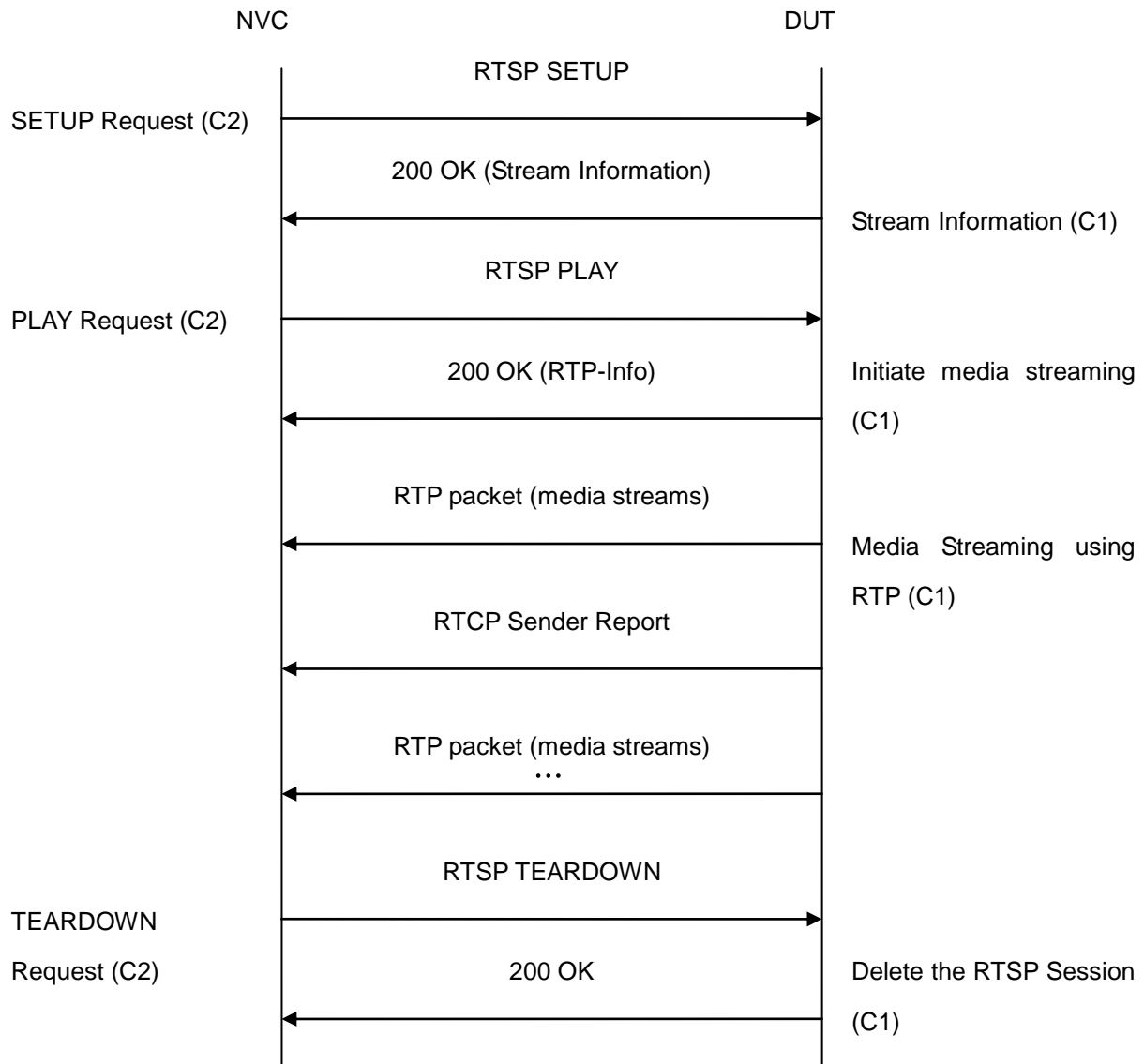
**Pre-Requisite:** A media profile with JPEG video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.

5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, HTTP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the HTTP media stream URI provided by the DUT.
9. NVC invokes HTTP GET Request on DUT and establishes DUT to NVC connection for RTP data transfer.
10. NVC invokes HTTP POST Request and establishes NVC to DUT connection for RTSP control requests.
11. NVC invokes RTSP DESCRIBE request on HTTP POST connection.
12. DUT sends 200 OK message and SDP information on HTTP GET connection.
13. NVC invokes RTSP SETUP request on HTTP POST connection with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter.
14. DUT sends 200 OK message and the media stream information on HTTP GET connection.
15. NVC invokes RTSP PLAY request on HTTP POST connection.
16. DUT sends 200 OK message and starts media streaming on HTTP GET connection.
17. DUT transfers JPEG RTP media stream to NVC on HTTP GET connection.
18. DUT sends RTCP sender report to NVC on HTTP GET connection.
19. DUT validates the received RTP and RTCP packets, decodes and renders them.
20. NVC invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.
21. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – HTTP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

#### **5.1.6 MEDIA STREAMING – JPEG (RTP/RTSP/TCP)**

**Test Label:** Real Time Viewing DUT JPEG media streaming using RTP/RTSP/TCP transport.

**Test Case ID:** MEDIA-1-1-6

**ONVIF Core Specification Coverage:** RTP/RTSP/TCP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (RTP/RTSP/TCP)

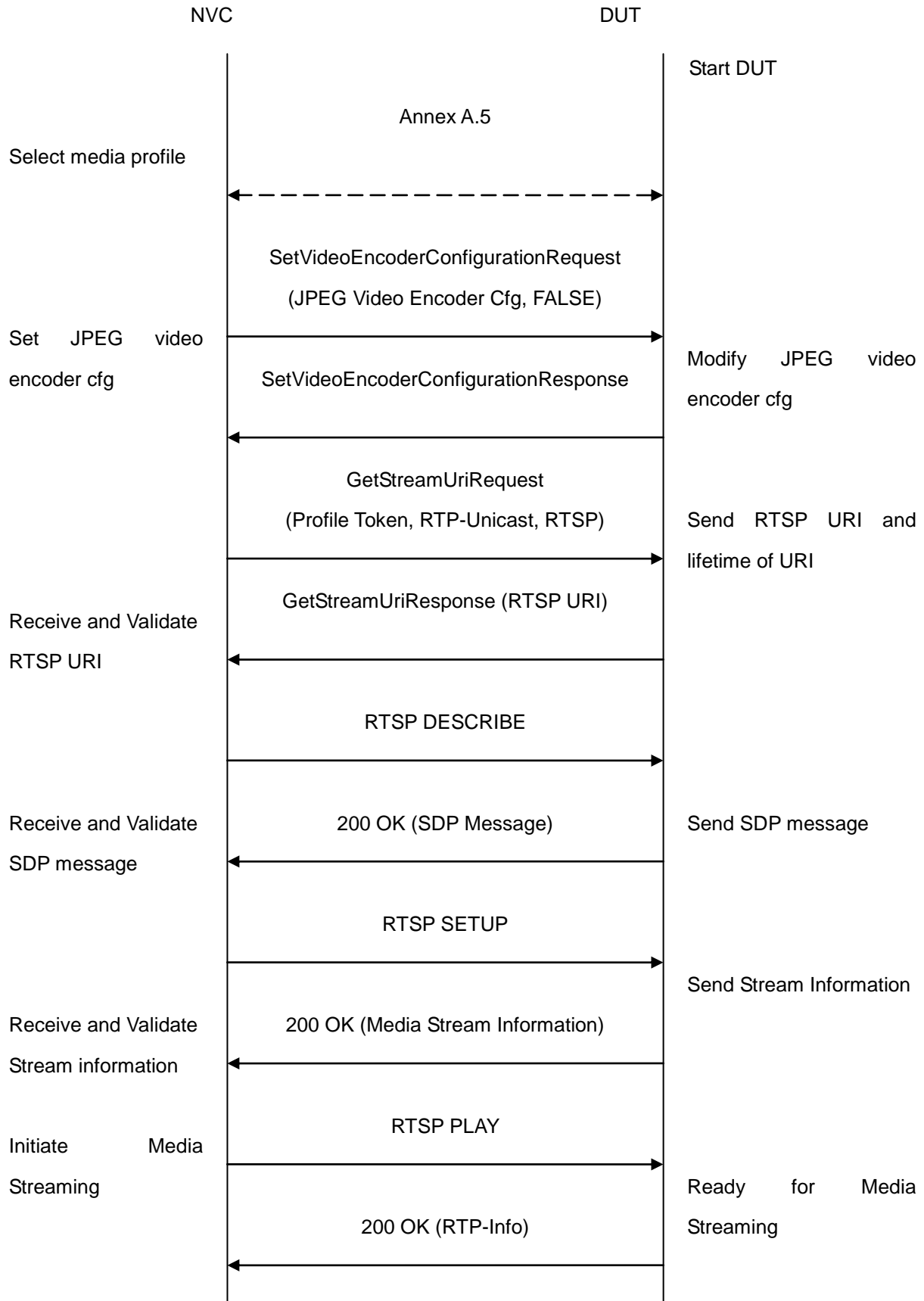
**Test Purpose:** To verify JPEG media streaming based on RTP/RTSP/TCP using RTSP tunnel.

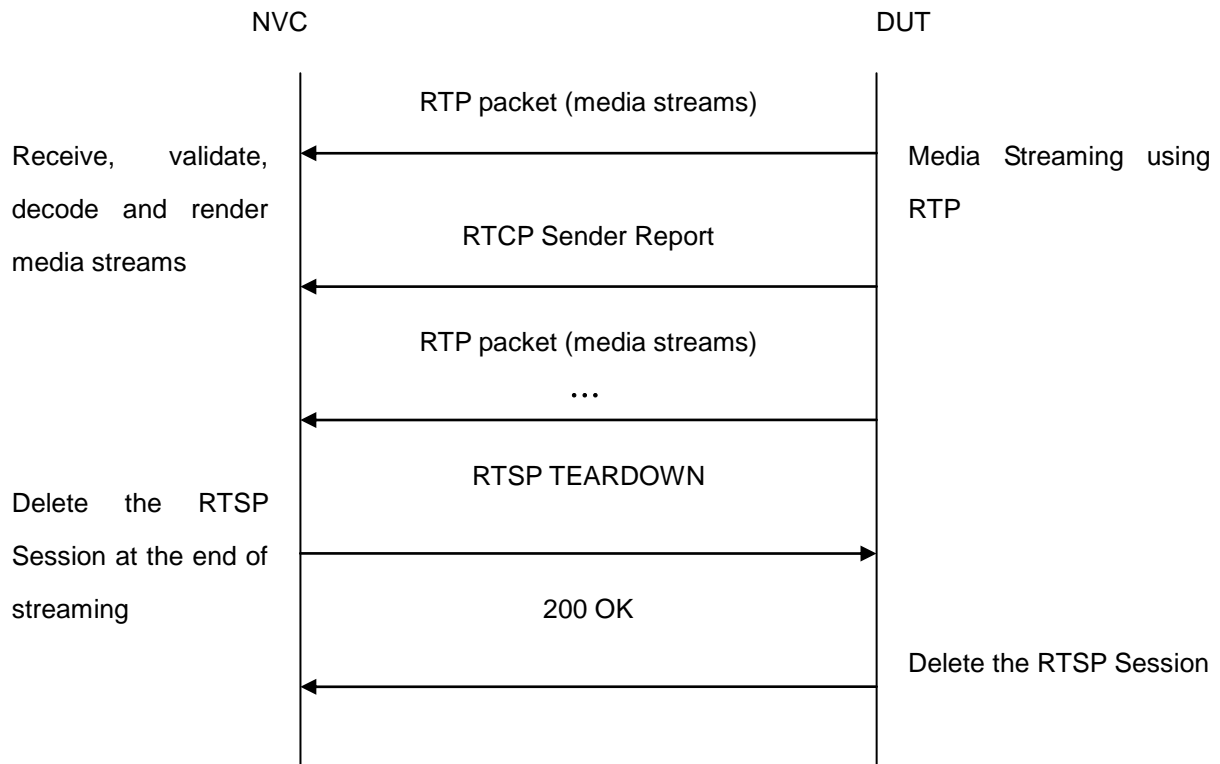
**Pre-Requisite:** RTP/RTSP/TCP media streaming is implemented by DUT.

A media profile with JPEG video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, RTSP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.

11. NVC invokes RTSP SETUP request with transport parameter as 'RTP/TCP' along with 'interleaved' parameter.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
16. DUT validates the received RTP and RTCP packets, decodes and renders them.
17. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
18. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

### 5.1.7 MEDIA STREAMING – MPEG4 (RTP-Unicast/ UDP)

**Test Label:** Real Time Viewing DUT MPEG4 media streaming using RTP-Unicast/UDP transport.

**Test Case ID:** RTSS-1-1-7

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (MPEG4-SP)

**Test Purpose:** To verify MPEG4 media streaming based on RTP/UDP Unicast Transport.

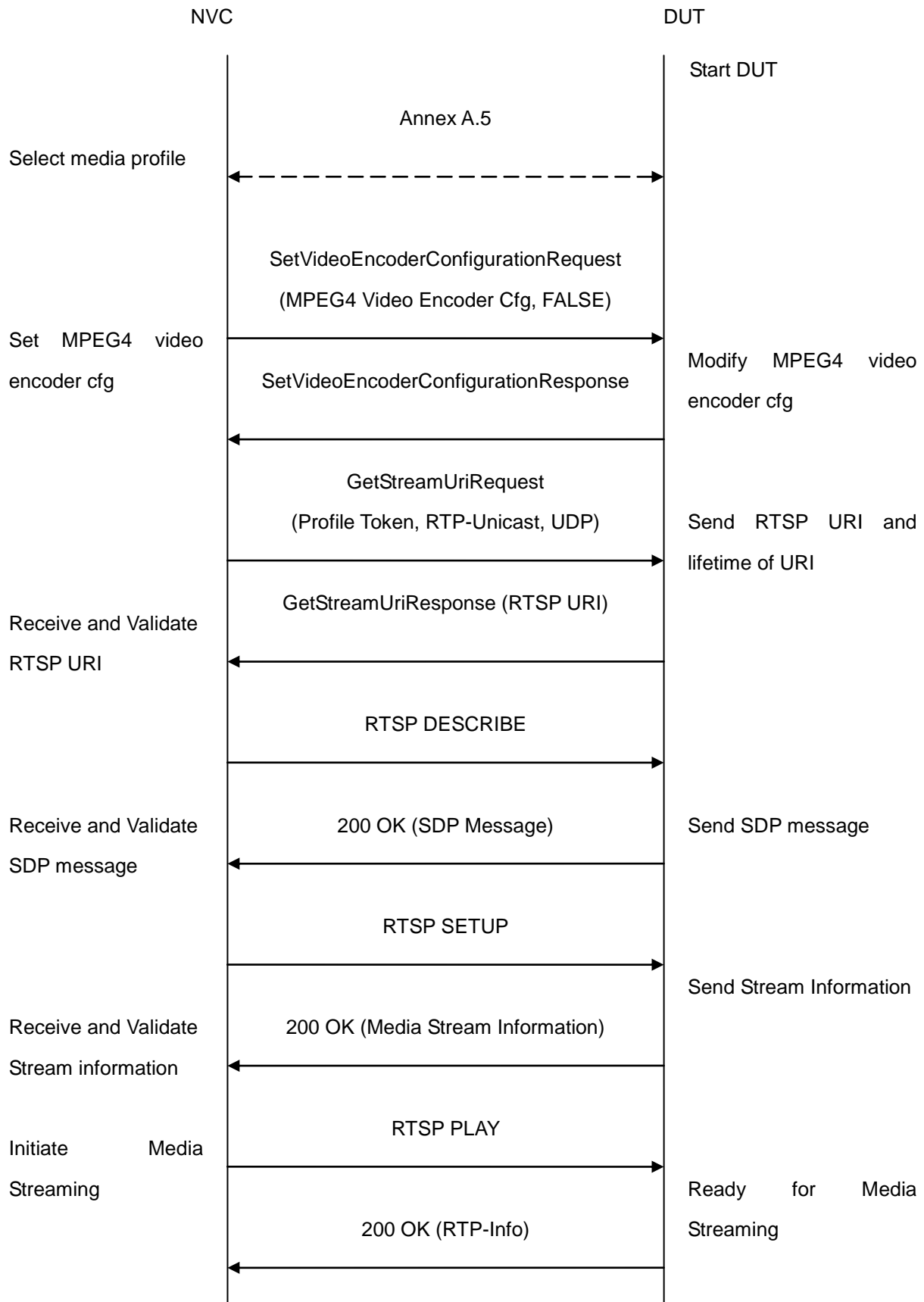
**Pre-Requisite:** MPEG4 is implemented by DUT

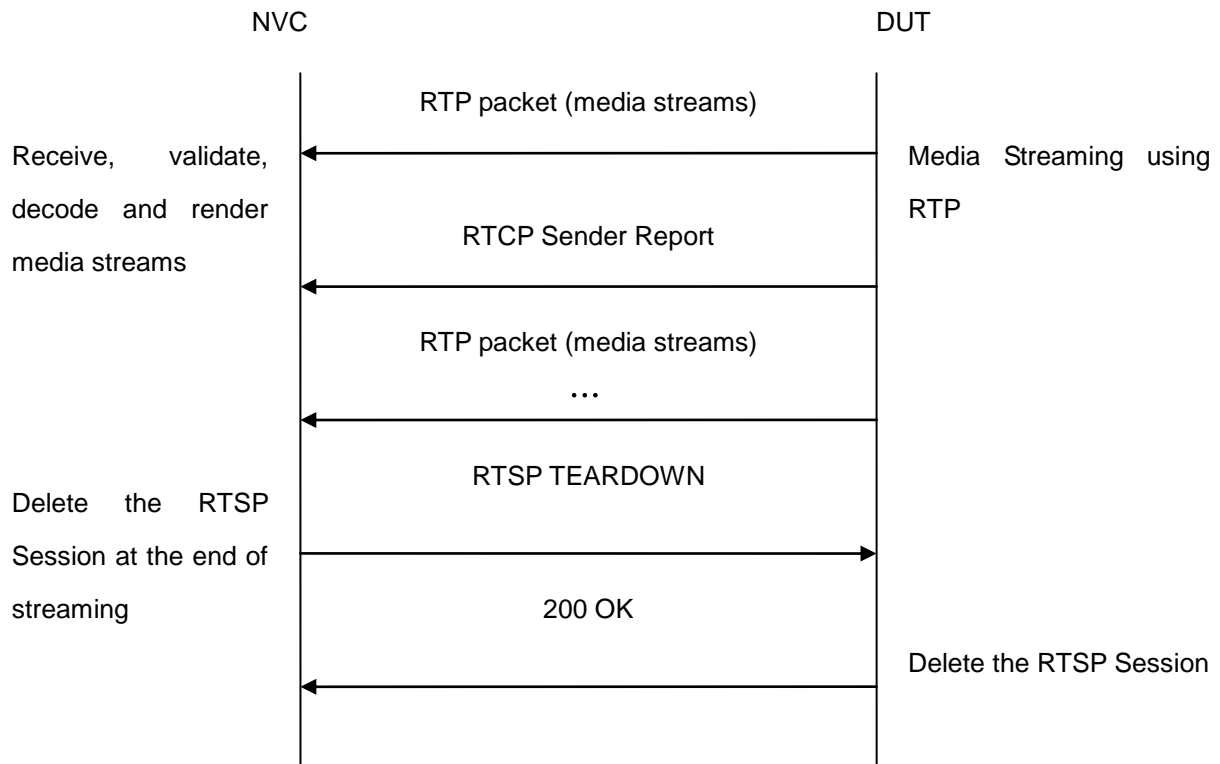
A media profile with MPEG4 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, Mpeg4Profile = SP, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.

11. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP**.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends MPEG4 RTP media stream to NVC over UDP.
16. DUT sends RTCP sender report to NVC.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

**5.1.8 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT MPEG4 media streaming using HTTP transport.

**Test Case ID:** RTSS-1-1-8

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (MPEG4-SP)

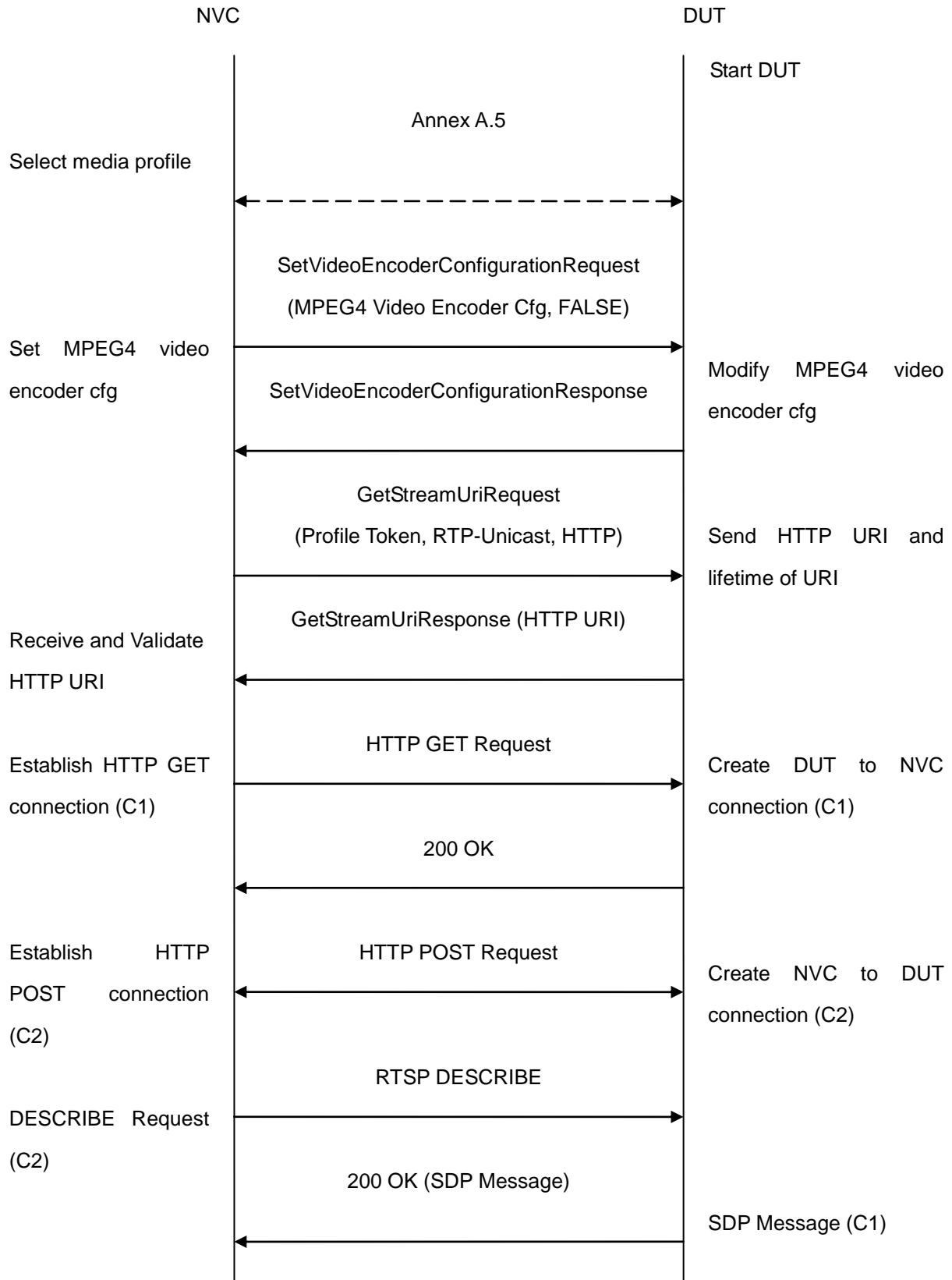
**Test Purpose:** To verify MPEG4 media streaming based on HTTP Transport.

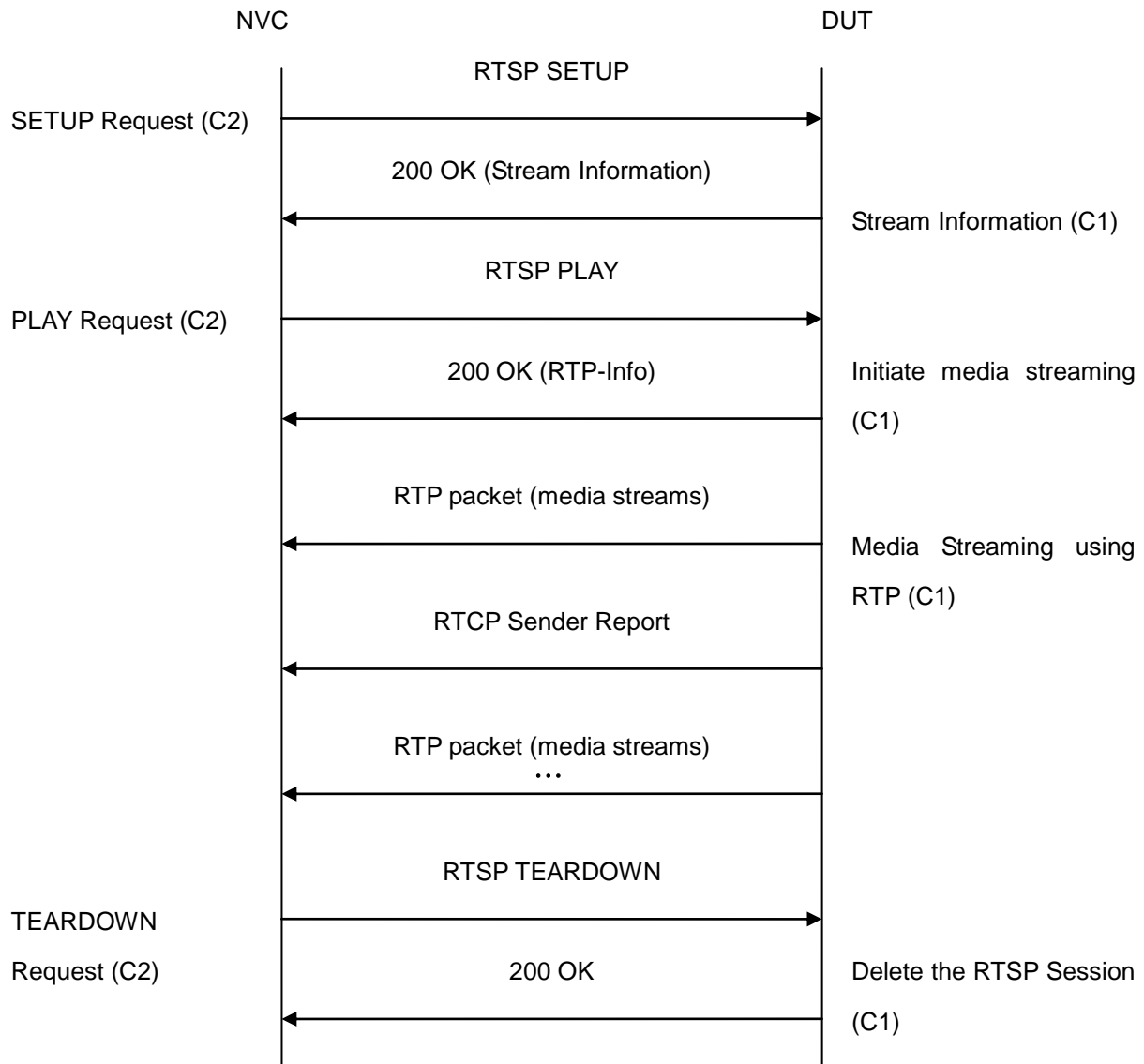
**Pre-Requisite:** MPEG4 is implemented by DUT.

A media profile with MPEG4 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes `SetVideoEncoderConfigurationRequest` (Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, Mpeg4Profile = SP, Session Timeout = t1 and force persistence = false). These values will be taken from the `GetVideoEncoderConfigurationOptions` response in A.5.
5. DUT modifies video encoder configuration and responds with `SetVideoEncoderConfigurationResponse` message indicating success.

6. NVC invokes GetStreamUriRequest message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the HTTP media stream URI provided by the DUT.
9. NVC invokes HTTP GET Request on DUT and establishes DUT to NVC connection for RTP data transfer.
10. NVC invokes HTTP POST Request and establishes NVC to DUT connection for RTSP control requests.
11. NVC invokes RTSP DESCRIBE request on HTTP POST connection.
12. DUT sends 200 OK message and SDP information on HTTP GET connection.
13. NVC invokes RTSP SETUP request on HTTP POST connection with transport parameter as 'RTP/TCP' along with 'interleaved' parameter.
14. DUT sends 200 OK message and the media stream information on HTTP GET connection.
15. NVC invokes RTSP PLAY request on HTTP POST connection.
16. DUT sends 200 OK message and starts media streaming on HTTP GET connection.
17. DUT transfers MPEG4 RTP media stream to NVC on HTTP GET connection.
18. DUT sends RTCP sender report to NVC on HTTP GET connection.
19. DUT validates the received RTP and RTCP packets, decodes and renders them.
20. NVC invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.
21. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – HTTP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.  
See Annex A.2 for Invalid RTP header definition.

#### 5.1.9 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP)

**Test Label:** Real Time Viewing DUT MPEG4 media streaming using RTP/RTSP/TCP transport.

**Test Case ID:** RTSS-1-1-9

**ONVIF Core Specification Coverage:** RTP/RTSP/TCP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (MPEG4-SP & RTP/RTSP/TCP)

**Test Purpose:** To verify MPEG4 media streaming based on RTP/RTSP/TCP using RTSP tunnel.

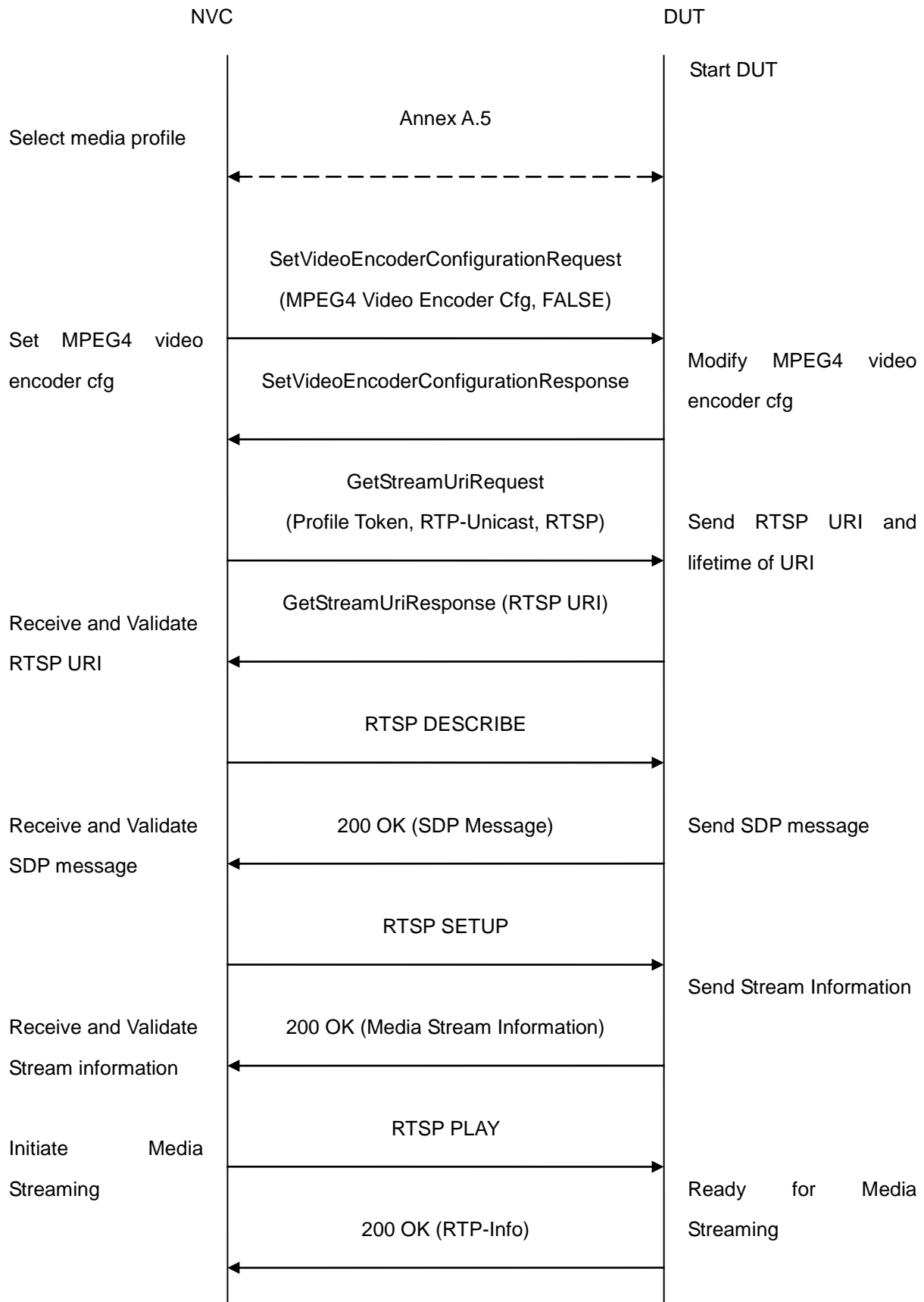
**Pre-Requisite:** MPEG4 and RTP/RTSP/TCP media streaming is implemented by DUT.

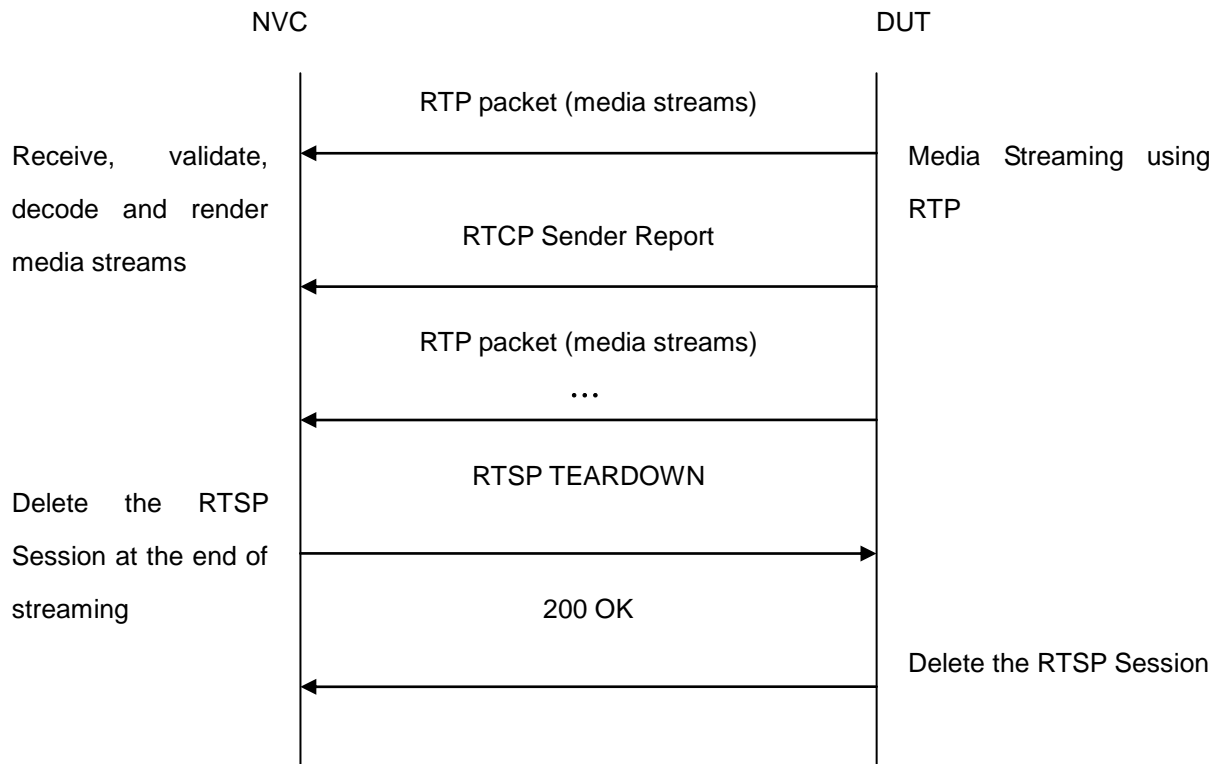
A media profile with MPEG4 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, Mpeg4Profile = SP, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, RTSP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.

11. NVC invokes RTSP SETUP request with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
16. DUT validates the received RTP and RTCP packets, decodes and renders them.
17. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
18. DUT sends 200 OK Response and terminates the RTSP Session.
- 19.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not send GetProfilesResponse message.

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

#### 5.1.10 SET SYNCHRONIZATION POINT – MPEG4

**Test Label:** Media Configuration DUT Synchronization Point – MPEG4

**Test Case ID:** RTSS-1-1-10

**ONVIF Core Specification Coverage:** Set synchronization point.

**Command Under Test:** SetSynchronizationPoint

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF IMPLEMENTED (MPEG4-SP)

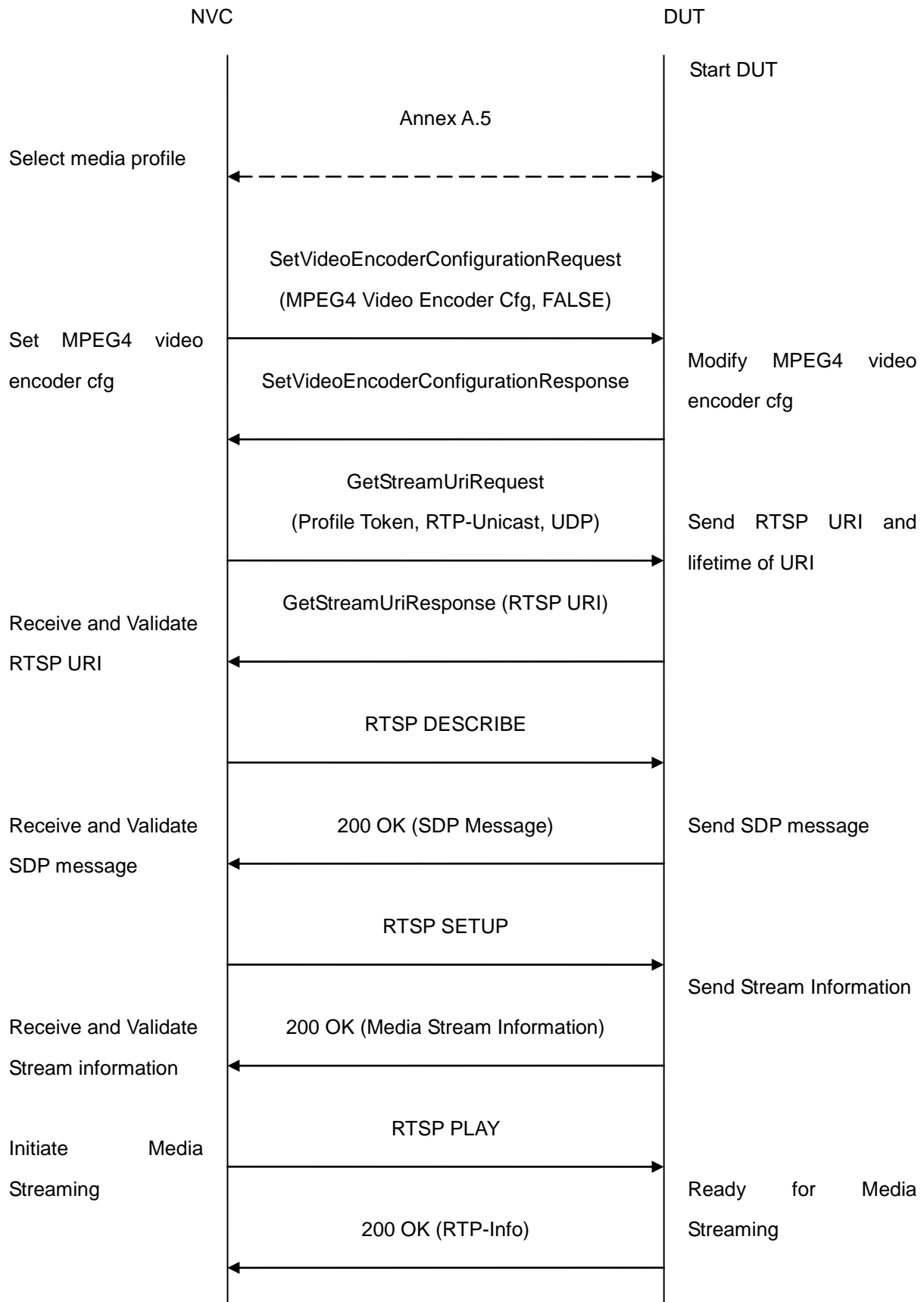
**Test Purpose:** To request synchronization point from DUT for MPEG4 media stream.

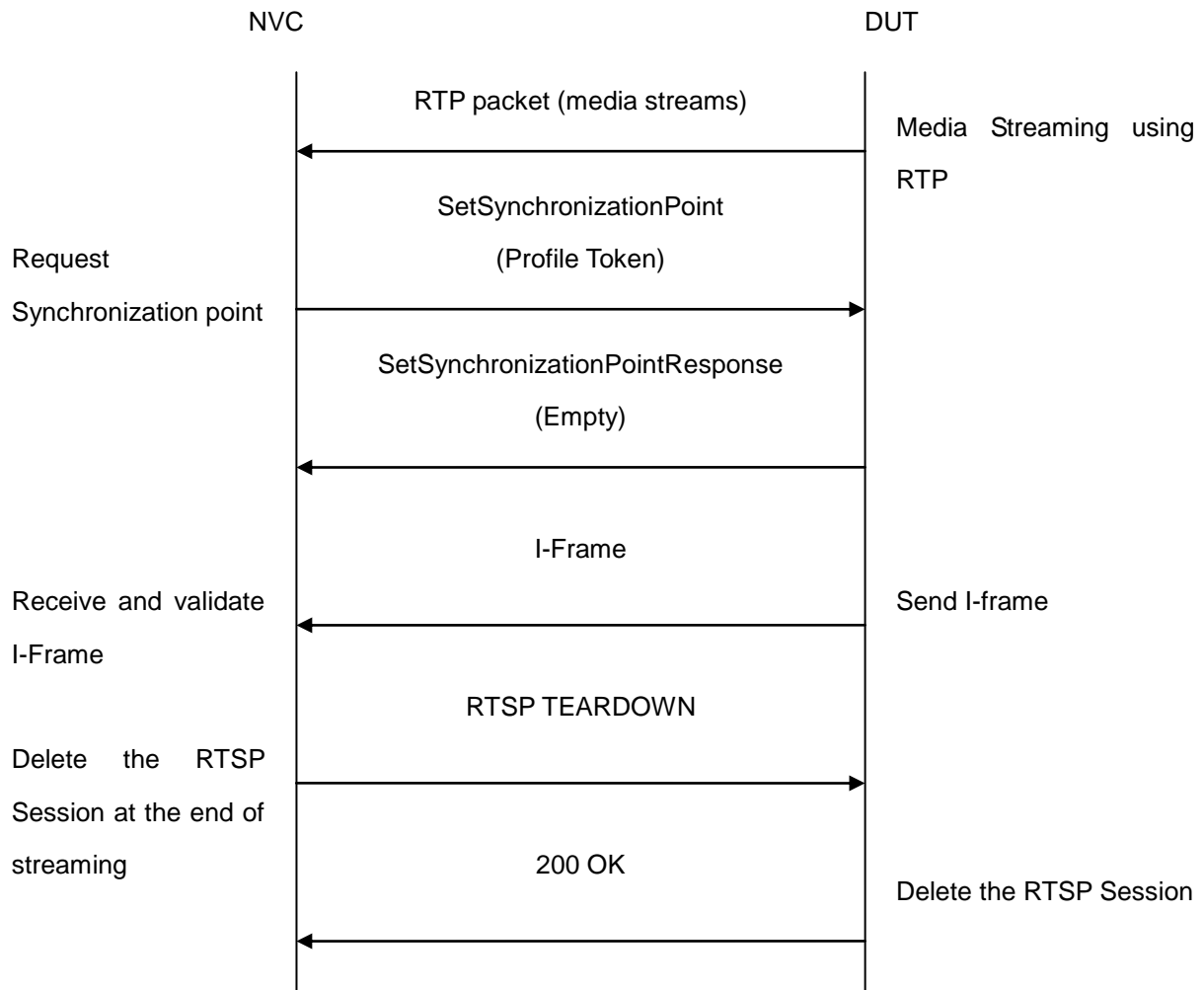
**Pre-Requisite:** MPEG4 is implemented by DUT.

A media profile with MPEG4 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, Mpeg4Profile = SP, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP**.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends MPEG4 RTP media stream to NVC over UDP.
16. DUT sends RTCP sender report to NVC.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes SetSynchronizationPoint request on the selected media profile.
19. DUT sends the SetSynchronizationPoint response indicating success.
20. DUT inserts the I-frame in the on going media stream.
21. NVC verifies that I-frame is sent by DUT before the regular 'I-frame insertion time interval'.
22. NVC invokes RTSP TEARDOWN control request to terminate the RTSP session.
23. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send SetSynchronizationPointResponse message.

DUT did not send I-frame before the regular 'I-frame insertion time interval' upon invoking SetSynchronizationPoint request.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.4 for details on 'I-frame insertion time interval'.

#### **5.1.11 MEDIA STREAMING – H.264 (RTP-Unicast/ UDP)**

**Test Label:** Real Time Viewing DUT H.264 media streaming using RTP-Unicast/UDP transport.

**Test Case ID:** RTSS-1-1-11

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (H.264-Baseline)

**Test Purpose:** To verify H.264 media streaming based on RTP/UDP Unicast Transport.

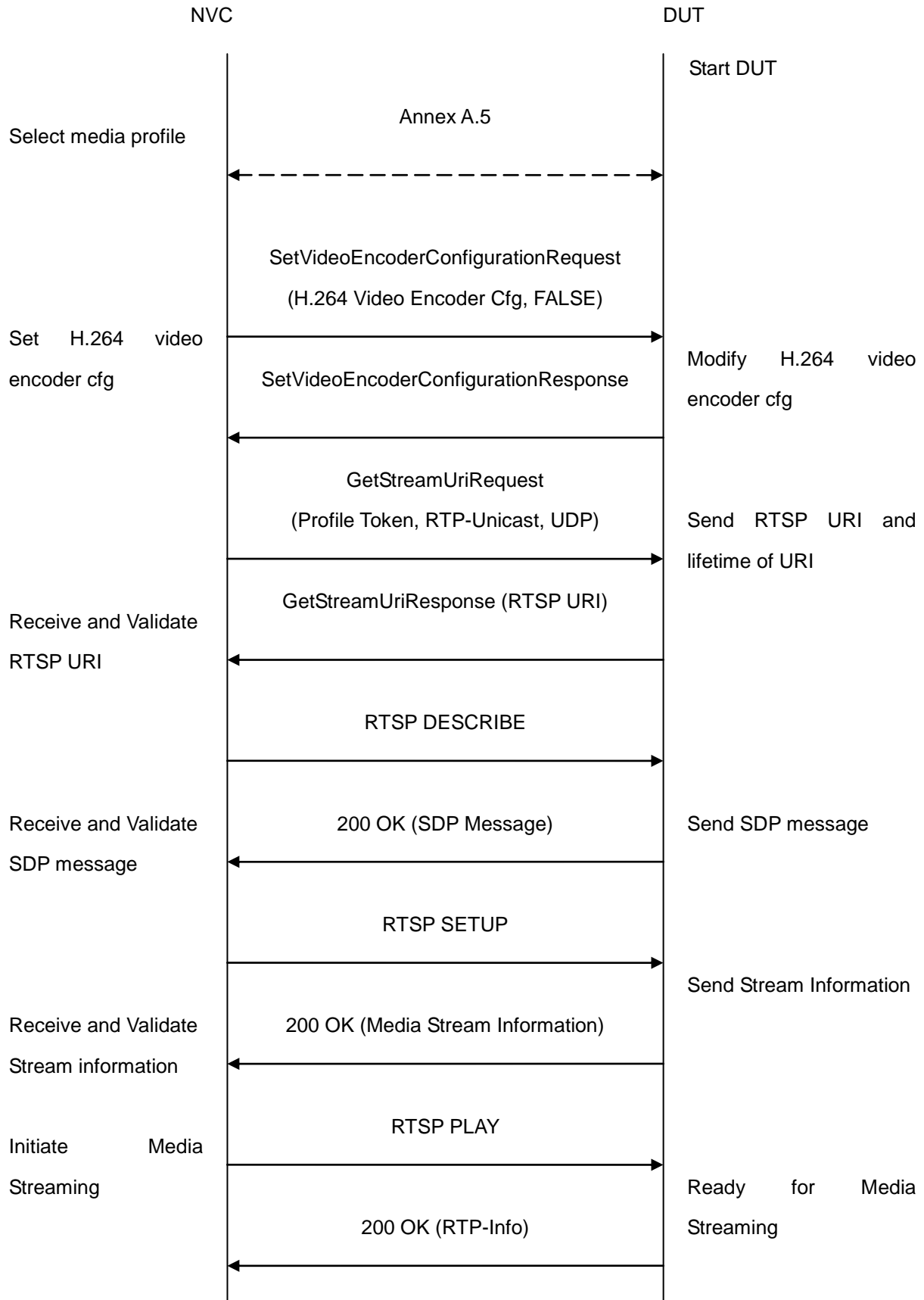
**Pre-Requisite:** H.264 is implemented by DUT

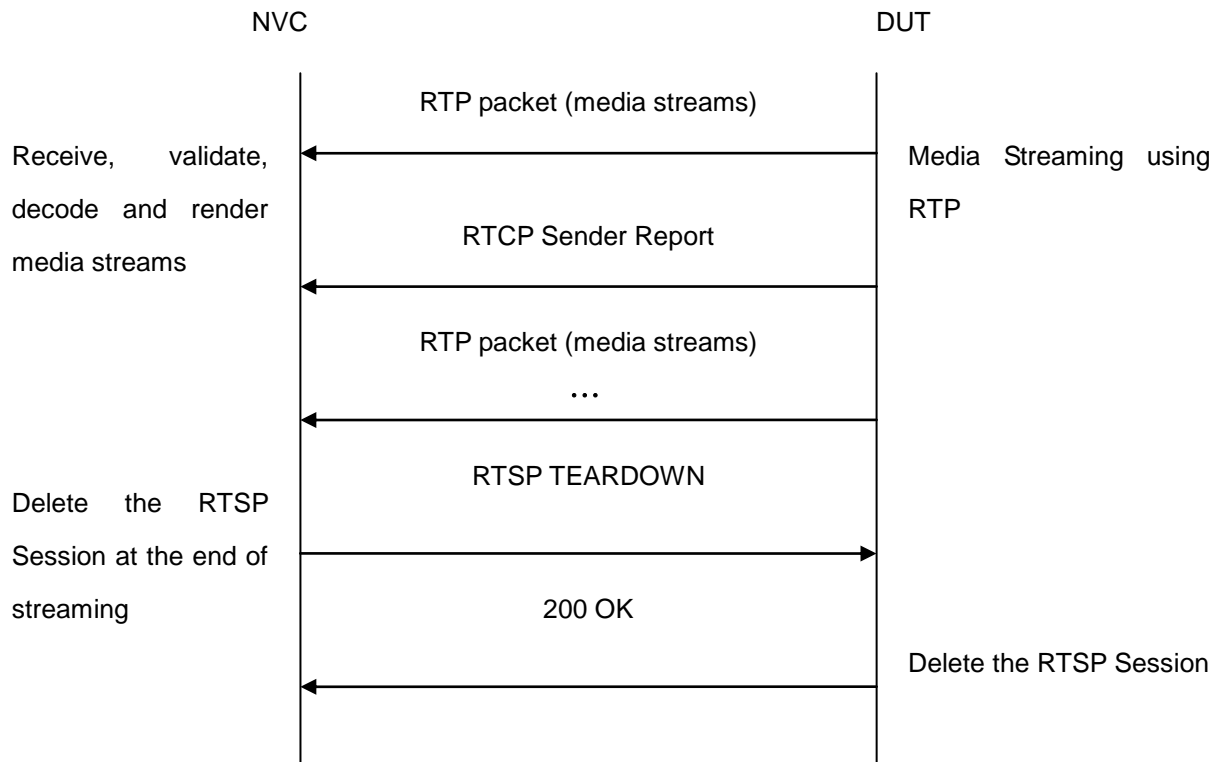
A media profile with H.264 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "H.264", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, H264Profile = Baseline, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.

11. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP**.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends H.264 RTP media stream to NVC over UDP.
16. DUT sends RTCP sender report to NVC.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

**5.1.12 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT H.264 media streaming using HTTP transport.

**Test Case ID:** RTSS-1-1-12



**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (H.264-Baseline)

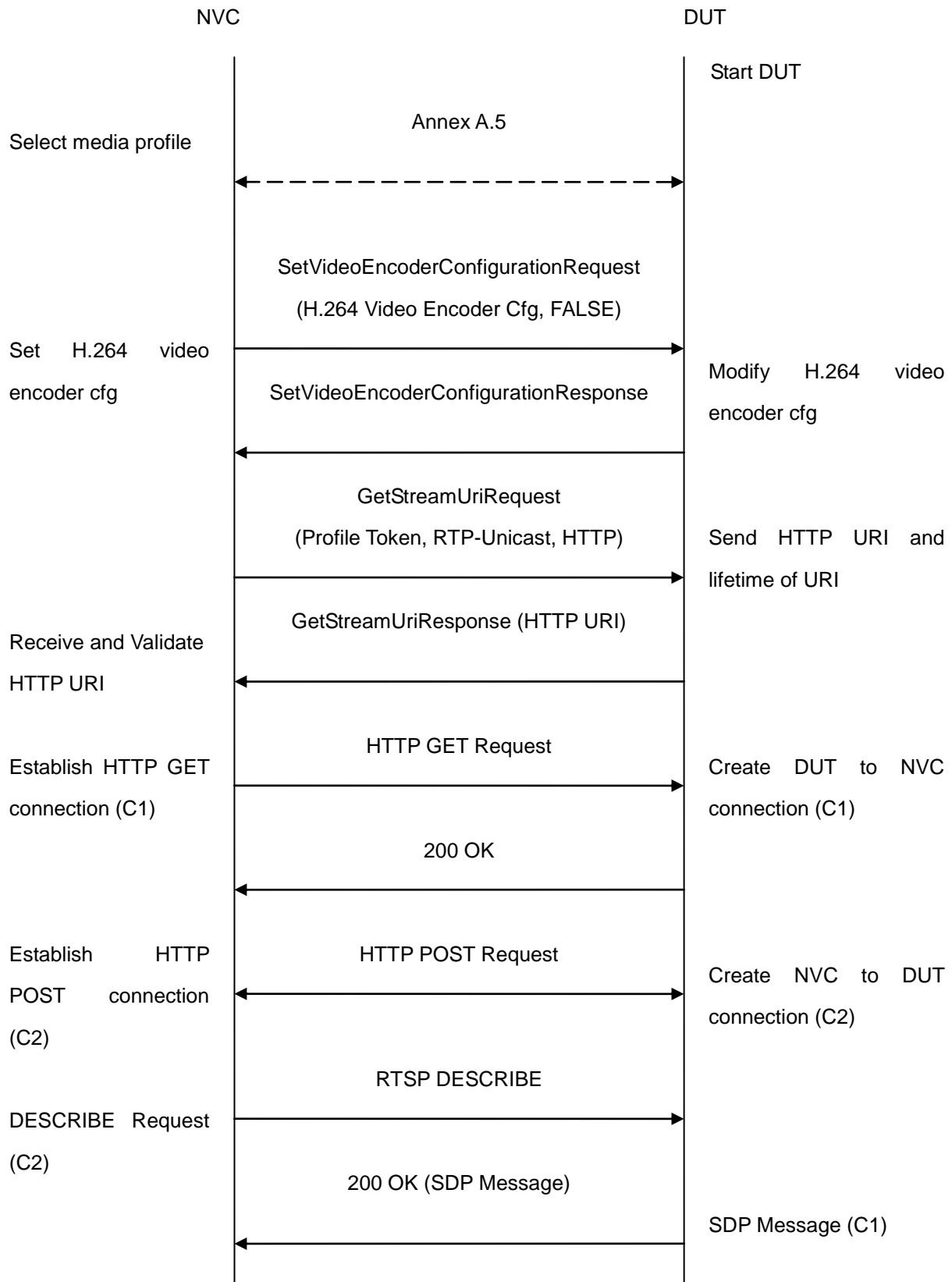
**Test Purpose:** To verify H.264 media streaming based on HTTP Transport.

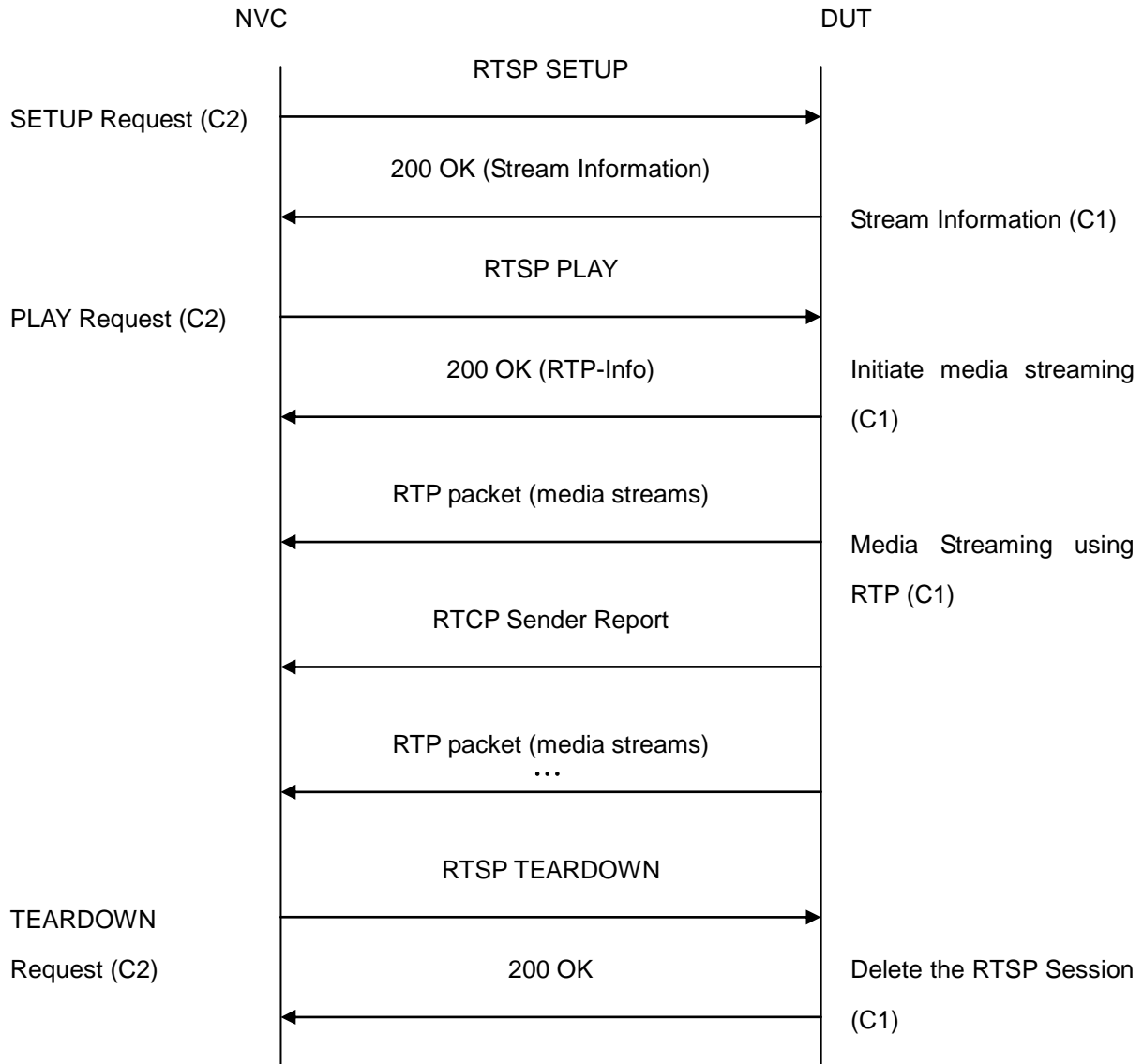
**Pre-Requisite:** H.264 is implemented by DUT.

A media profile with H.264 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.5.

NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "H.264", Resolution = ["Width", "Height"])  
**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

1. See Annex A.2 for Invalid RTP header definition.gh”], Quality = q1, GovLength = g1, H264Profile = Baseline, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
2. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
3. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, HTTP transport**) to retrieve media stream URI for the selected media profile.
4. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
5. NVC verifies the HTTP media stream URI provided by the DUT.
6. NVC invokes HTTP GET Request on DUT and establishes DUT to NVC connection for RTP data transfer.
7. NVC invokes HTTP POST Request and establishes NVC to DUT connection for RTSP control requests.
8. NVC invokes RTSP DESCRIBE request on HTTP POST connection.
9. DUT sends 200 OK message and SDP information on HTTP GET connection.
10. NVC invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘**RTP/TCP**’ along with ‘**interleaved**’ parameter.
11. DUT sends 200 OK message and the media stream information on HTTP GET connection.
12. NVC invokes RTSP PLAY request on HTTP POST connection.
13. DUT sends 200 OK message and starts media streaming on HTTP GET connection.
14. DUT transfers H.264 RTP media stream to NVC on HTTP GET connection.
15. DUT sends RTCP sender report to NVC on HTTP GET connection.
16. DUT validates the received RTP and RTCP packets, decodes and renders them.
17. NVC invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.
18. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.
- 19.

#### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – HTTP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.1.13 MEDIA STREAMING – H.264 (RTP/RTSP/TCP)

**Test Label:** Real Time Viewing DUT H.264 media streaming using RTP/RTSP/TCP transport.

**Test Case ID:** RTSS-1-1-13

**ONVIF Core Specification Coverage:** RTP/RTSP/TCP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF IMPLEMENTED (H.264-Baseline & RTP/RTSP/TCP)

**Test Purpose:** To verify H.264 media streaming based on RTP/RTSP/TCP using RTSP tunnel.

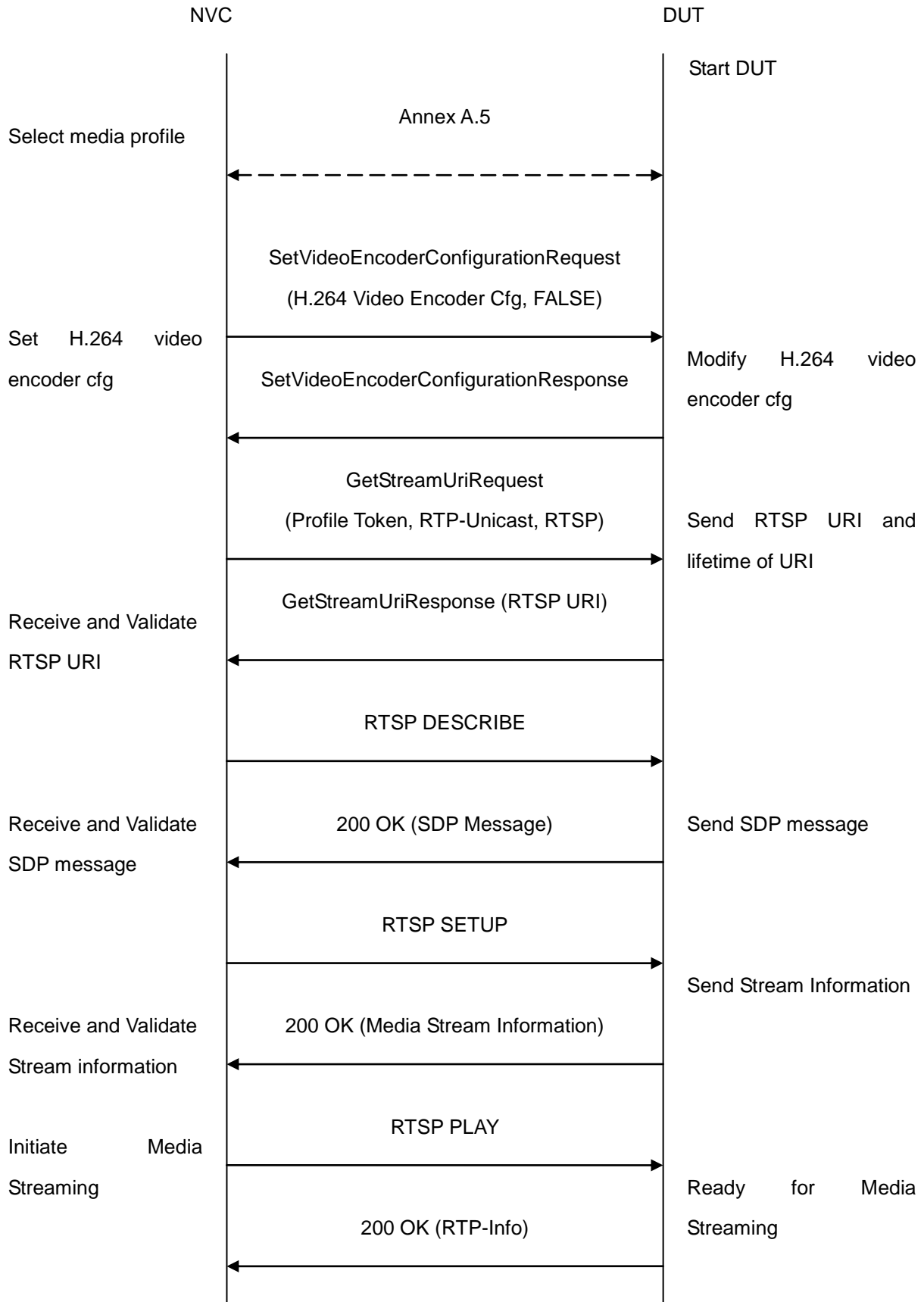
**Pre-Requisite:** H.264 and RTP/RTSP/TCP media streaming is implemented by DUT.

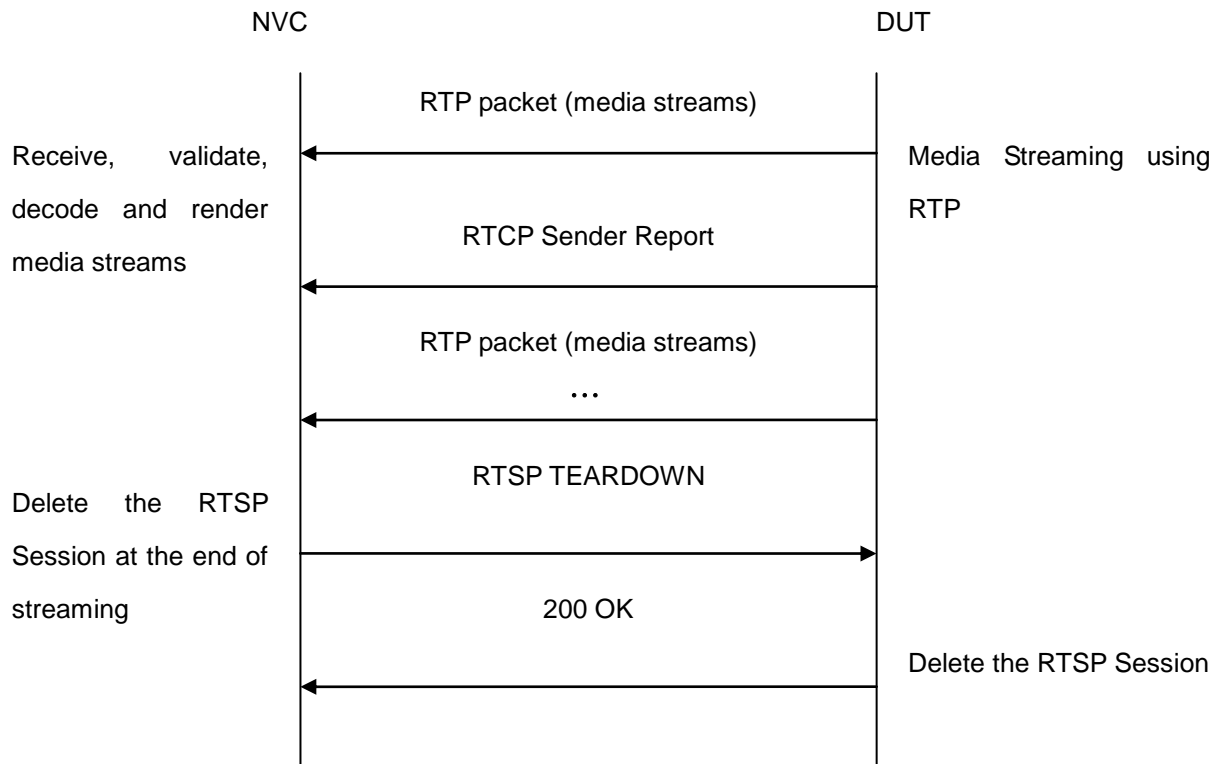
A media profile with H.264 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "H.264", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, H264Profile = Baseline, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.5.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, RTSP transport**) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.

11. NVC invokes RTSP SETUP request with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
16. DUT validates the received RTP and RTCP packets, decodes and renders them.
17. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
18. DUT sends 200 OK Response and terminates the RTSP Session.
- 19.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

#### **5.1.14 SET SYNCHRONIZATION POINT – H.264**

**Test Label:** Media Configuration DUT Synchronization Point – H.264

**Test Case ID:** RTSS-1-1-14

**ONVIF Core Specification Coverage:** Set synchronization point.

**Command Under Test:** SetSynchronizationPoint

**WSDL Reference:** media.wsdl

**Requirement Level:** MUST IF IMPLEMENTED (H.264-Baseline)

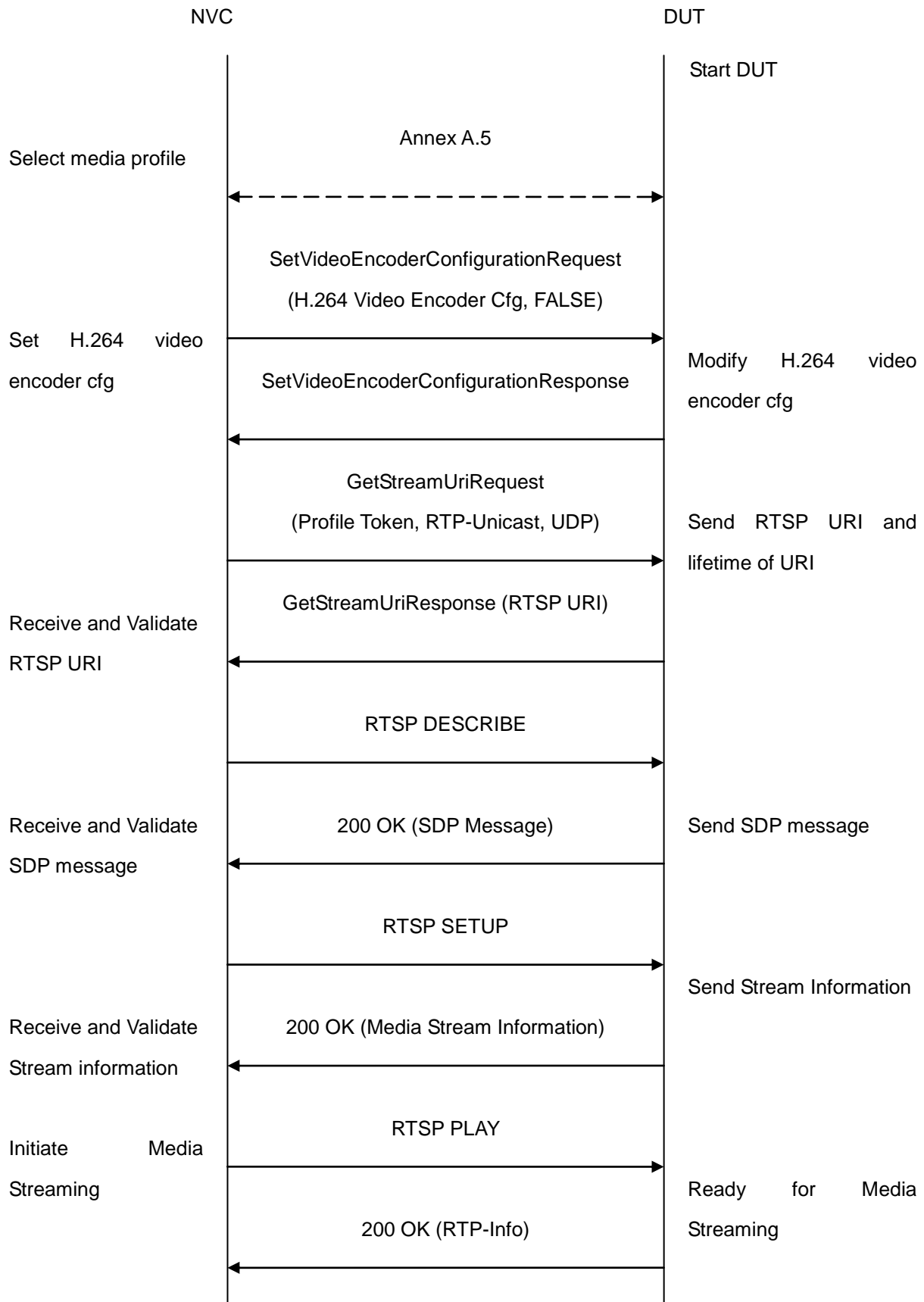
**Test Purpose:** To request synchronization point from DUT for H.264 media stream.

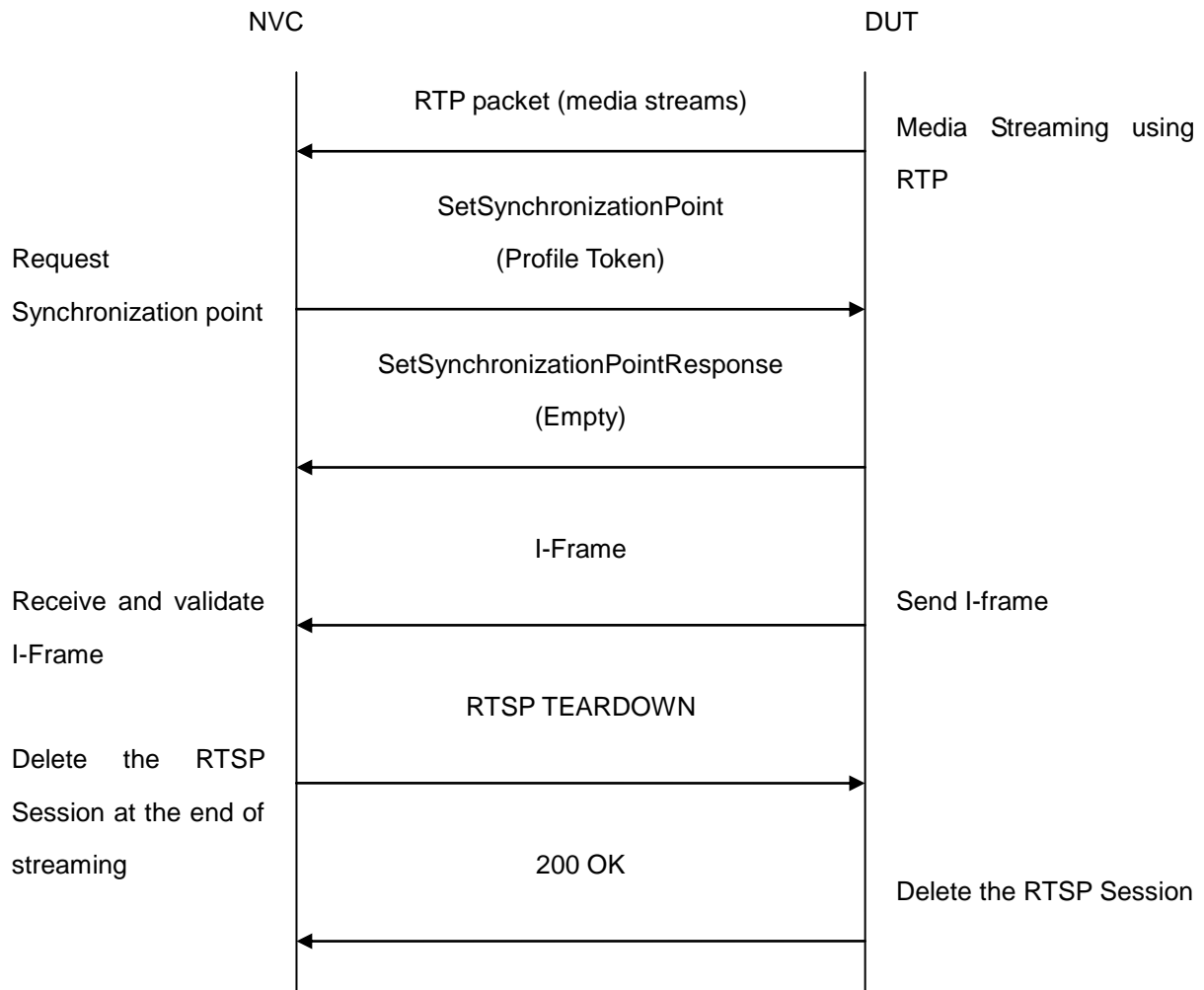
**Pre-Requisite:** H.264 is implemented by DUT.

A media profile with H.264 video encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.5.
4. NVC invokes **SetVideoEncoderConfigurationRequest** (Encoding = "H.264", Resolution = ["Width", "Height"], Quality = q1, GovLength = g1, H264Profile = Baseline, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptions** response in A.5.
5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. NVC invokes **GetStreamUriRequest** message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP**.
12. DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends H.264 RTP media stream to NVC over UDP.
16. DUT sends RTCP sender report to NVC.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes SetSynchronizationPoint request on the selected media profile.
19. DUT sends the SetSynchronizationPointResponse indicating success.
20. DUT inserts the I-frame in the on going media stream.
21. NVC verifies that I-frame is sent by DUT before the regular 'I-frame insertion time interval'.
22. NVC invokes RTSP TEARDOWN control request to terminate the RTSP session.
23. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send SetSynchronizationPointResponse message.

DUT did not send I-frame before the regular 'I-frame insertion time interval' upon invoking SetSynchronizationPoint request.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.4 for details on 'I-frame insertion time interval'.

#### **5.1.15 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4)**

**Test Label:** Real Time Viewing DUT JPEG Media Streaming Using RTP-Multicast/UDP Transport for IPv4.

**Test Case ID:** RTSS-1-2-1

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, JPEG over RTP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** SHOULD

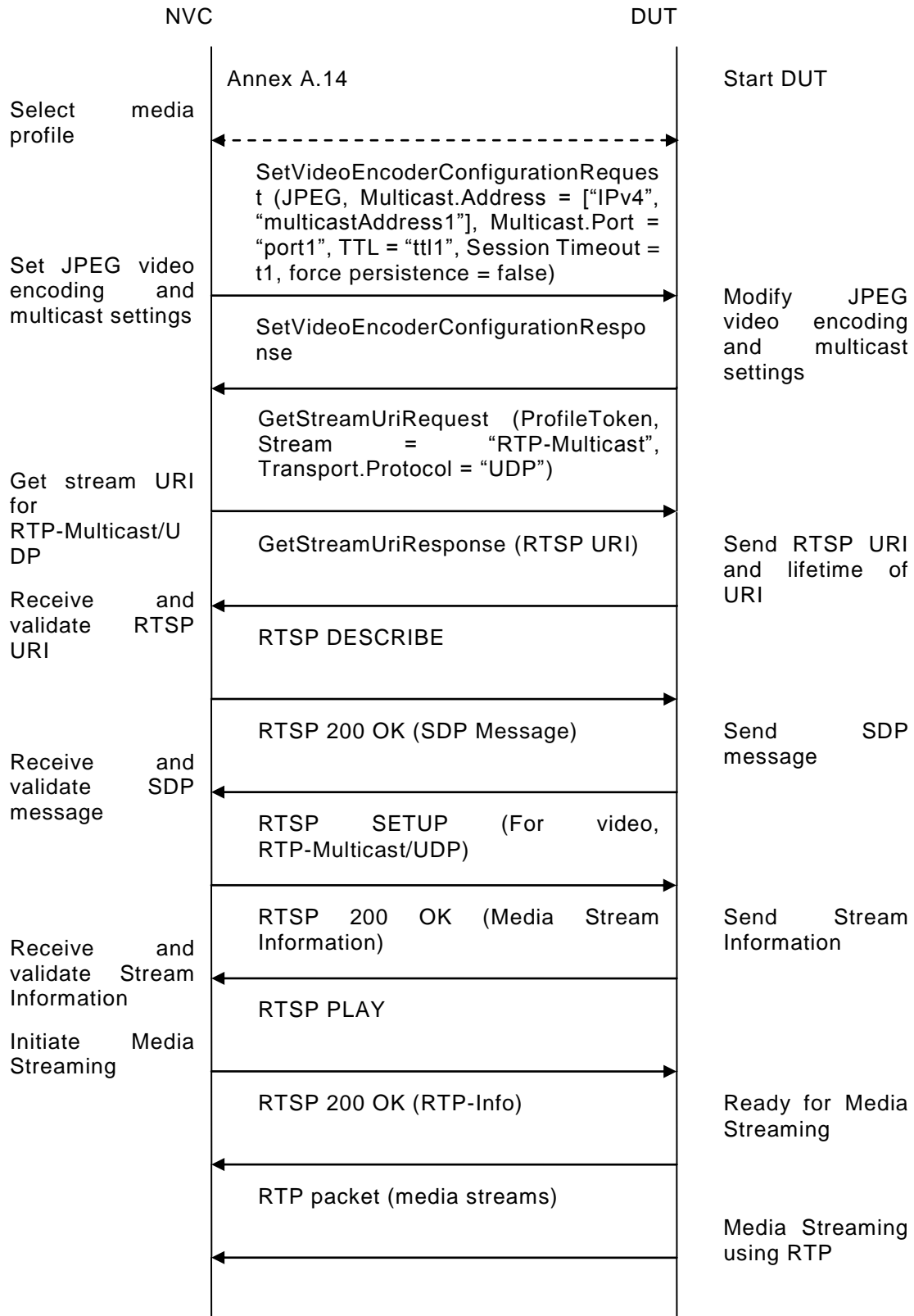
**Test Propose:** To verify JPEG media streaming based on RTP-Multicast/UDP Transport for IPv4.

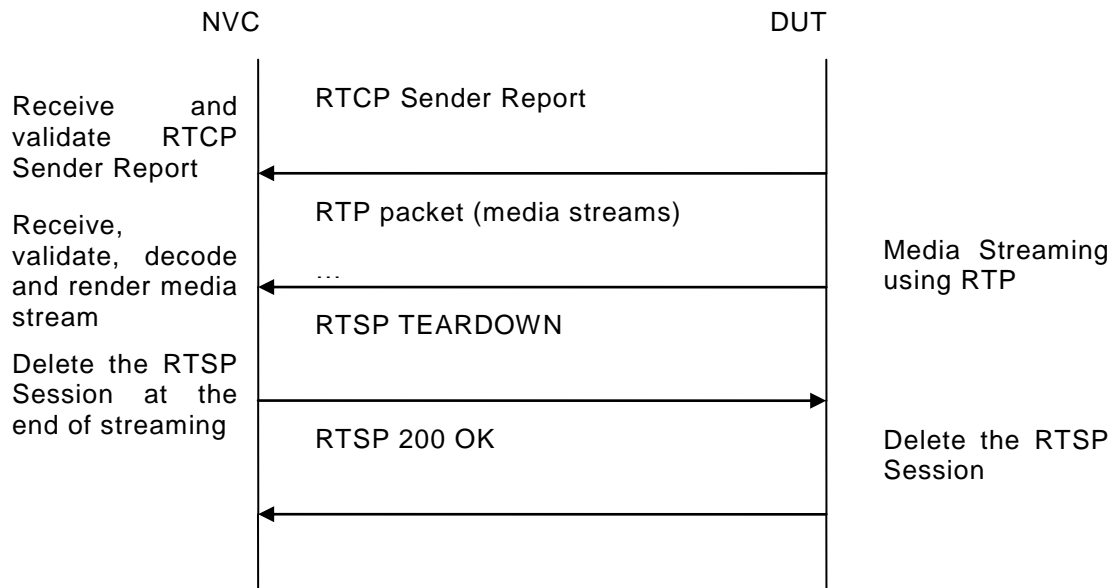
**Pre-Requisite:** Media is supported by DUT. A media profile with JPEG video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**

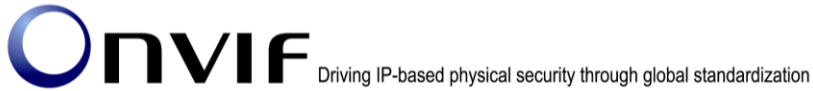






#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with JPEG video encoding support by the following procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest message (Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Multicast.Address = ["IPv4", "multicastAddress1"], Multicast.Port = "port1", TTL = "ttl1", Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings.
5. Verify the SetVideoEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream = "RTP-Multicast"**, **Transport.Protocol = "UDP"**) to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Multicast/UDP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends JPEG RTP multicast media stream to multicast IPv4 address over UDP.



16. The DUT sends RTCP sender report to NVC.
17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

#### **Test Result:**

##### **PASS –**

DUT passes all assertions.

##### **FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetVideoEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send JPEG RTP multicast media streaming for corresponding multicast IP.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

#### **5.1.16 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4)**

**Test Label:** Real Time Viewing DUT MPEG4 Media Streaming Using RTP-Multicast/UDP Transport for IPv4.

**Test Case ID:** RTSS-1-2-2

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

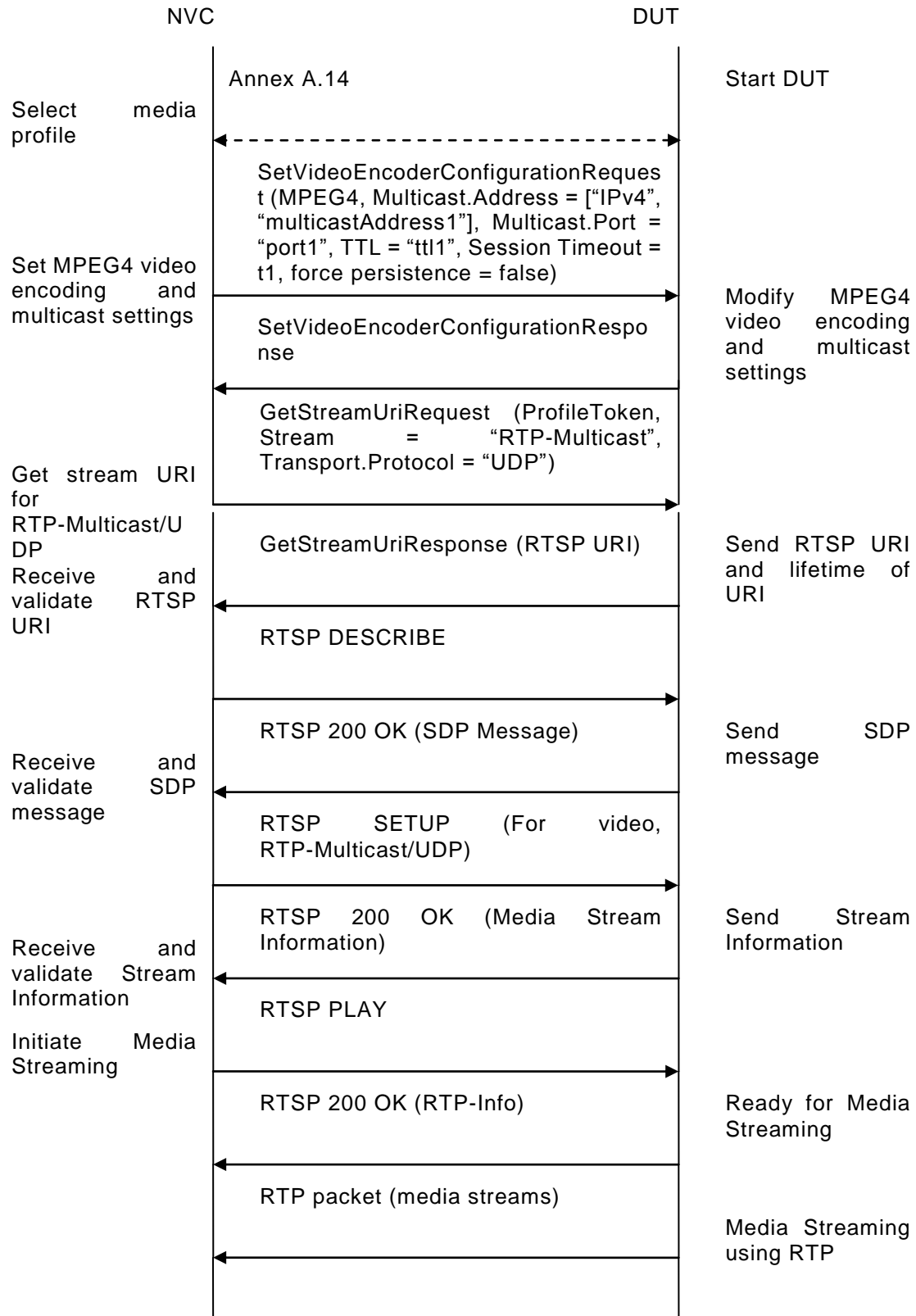
**Requirement Level:** SHOULD IF IMPLEMENTED (MPEG4-SP)

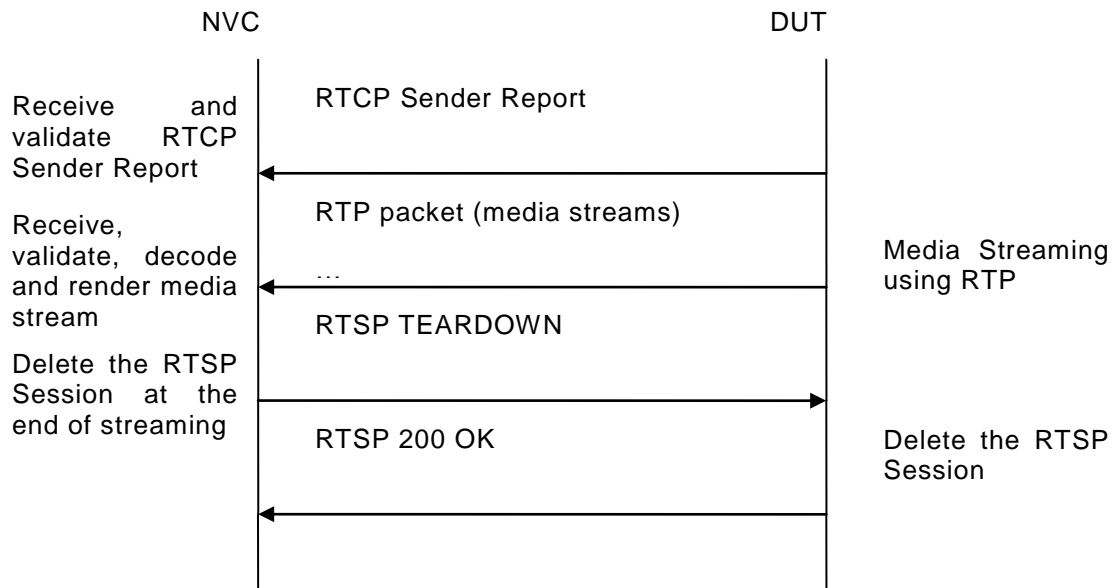
**Test Propose:** To verify MPEG4 media streaming based on RTP-Multicast/UDP Transport for IPv4.

**Pre-Requisite:** Media is supported by DUT and MPEG4 is implemented by DUT. A media profile with MPEG4 video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with MPEG4 video encoding support by the following procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest message (Encoding = "MPEG4", Resolution = ["Width", "Height"], Quality = q1, Multicast.Address = ["IPv4", "multicastAddress1"], Multicast.Port = "port1", TTL = "ttl1", Session Timeout = t1 and force persistence = false) to set MPEG4 encoding and Multicast settings.
5. Verify the SetVideoEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream = "RTP-Multicast"**, **Transport.Protocol = "UDP"**) to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Multicast/UDP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends MPEG4 RTP multicast media stream to multicast IPv4 address over UDP.

16. The DUT sends RTCP sender report to NVC.
17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetVideoEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send MPEG4 RTP multicast media streaming for corresponding multicast IP.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

**5.1.17 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4)**

**Test Label:** Real Time Viewing DUT H.264 Media Streaming Using RTP-Multicast/UDP Transport for IPv4.

**Test Case ID:** RTSS-1-2-3

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** SHOULD IF IMPLEMENTED (H.264-Baseline)

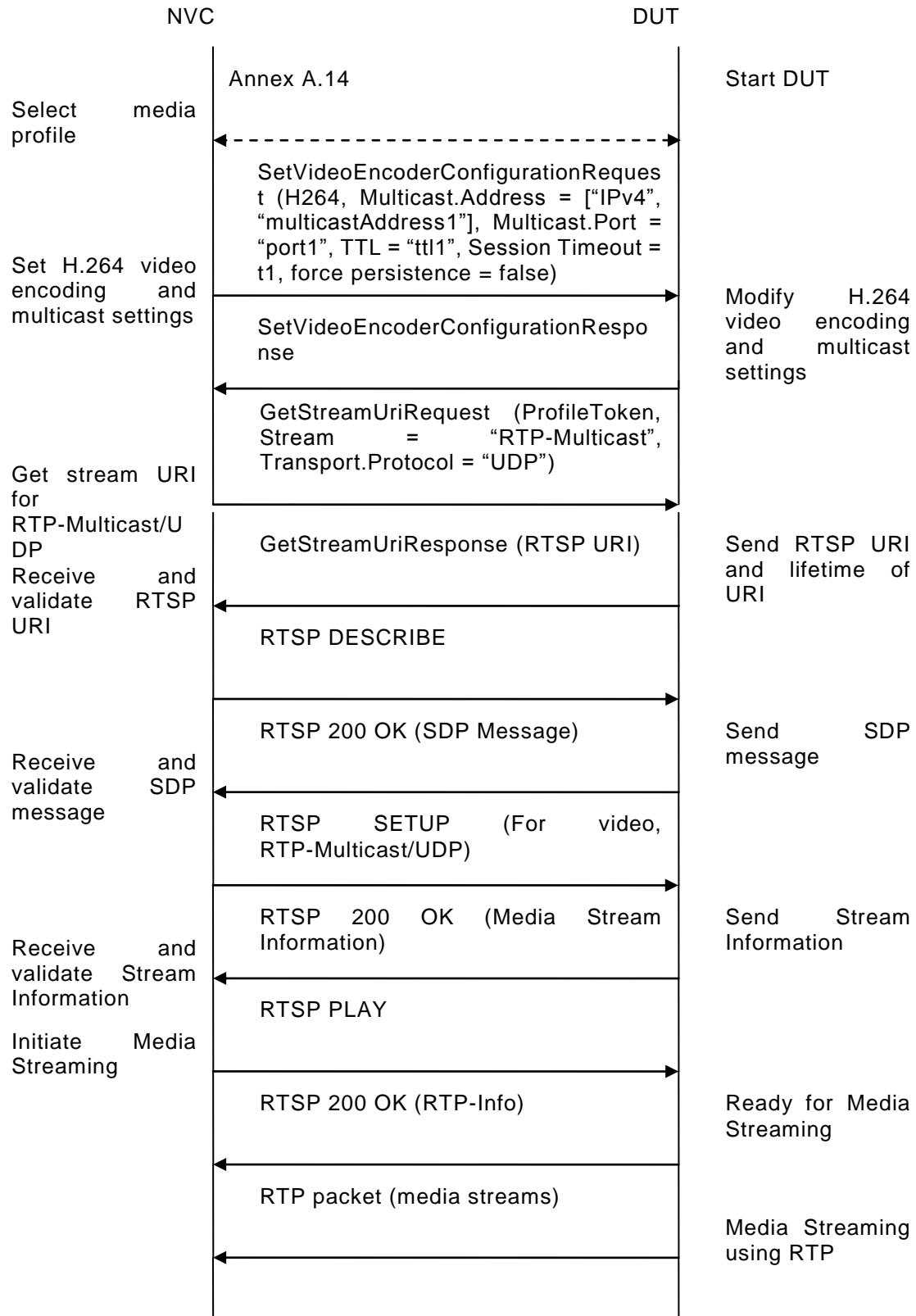
**Test Propose:** To verify H.264 media **streaming** based on RTP-Multicast/UDP Transport for **IPv4**.

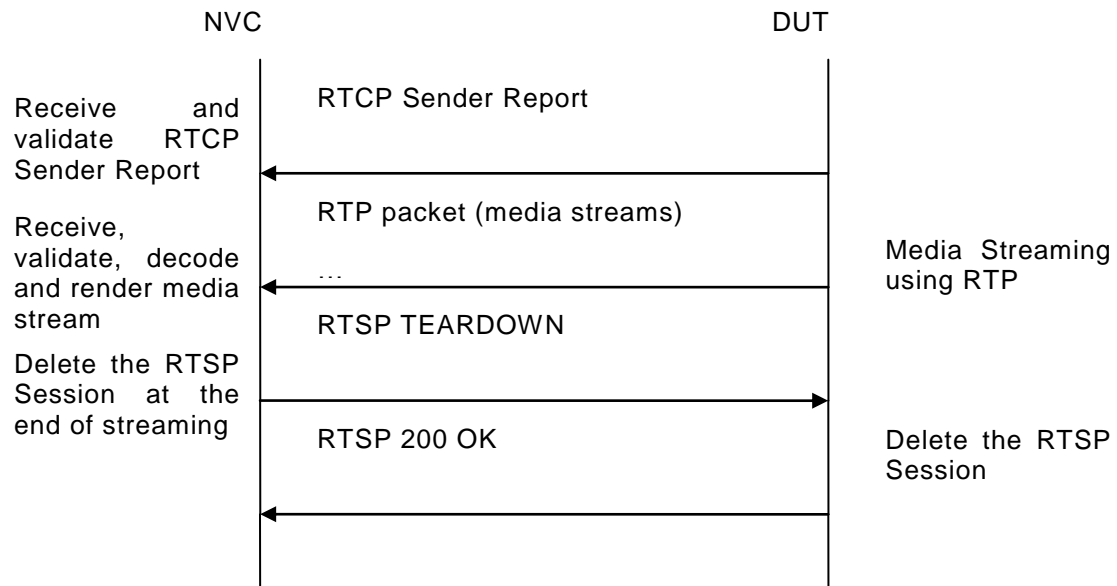
**Pre-Requisite:** **Media is supported by DUT** and **H.264 is implemented by DUT**. A media profile with H.264 video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with H.264 video encoding support by the following procedure mentioned in Annex A.5.
4. NVC invokes SetVideoEncoderConfigurationRequest message (Encoding = "H264", Resolution = ["Width", "Height"], Quality = q1, Multicast.Address = ["IPv4", "multicastAddress1"], Multicast.Port = "port1", TTL = "ttl1", Session Timeout = t1 and force persistence = false) to set H.264 encoding and Multicast settings.
5. Verify the SetVideoEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream = "RTP-Multicast"**, **Transport.Protocol = "UDP"**) to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Multicast/UDP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends H.264 RTP multicast media stream to multicast IPv4 address over UDP.

16. The DUT sends RTCP sender report to NVC.
17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetVideoEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send H.264 RTP multicast media streaming for corresponding multicast IP.

**Note:** See Annex A.5 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

## 5.2 Audio Streaming

### 5.2.1 MEDIA STREAMING – G.711 (RTP-Unicast/UDP)

**Test Label:** Real Time Viewing DUT G.711 Media Streaming Using RTP-Unicast/UDP Transport.

**Test Case ID:** RTSS-2-1-1

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

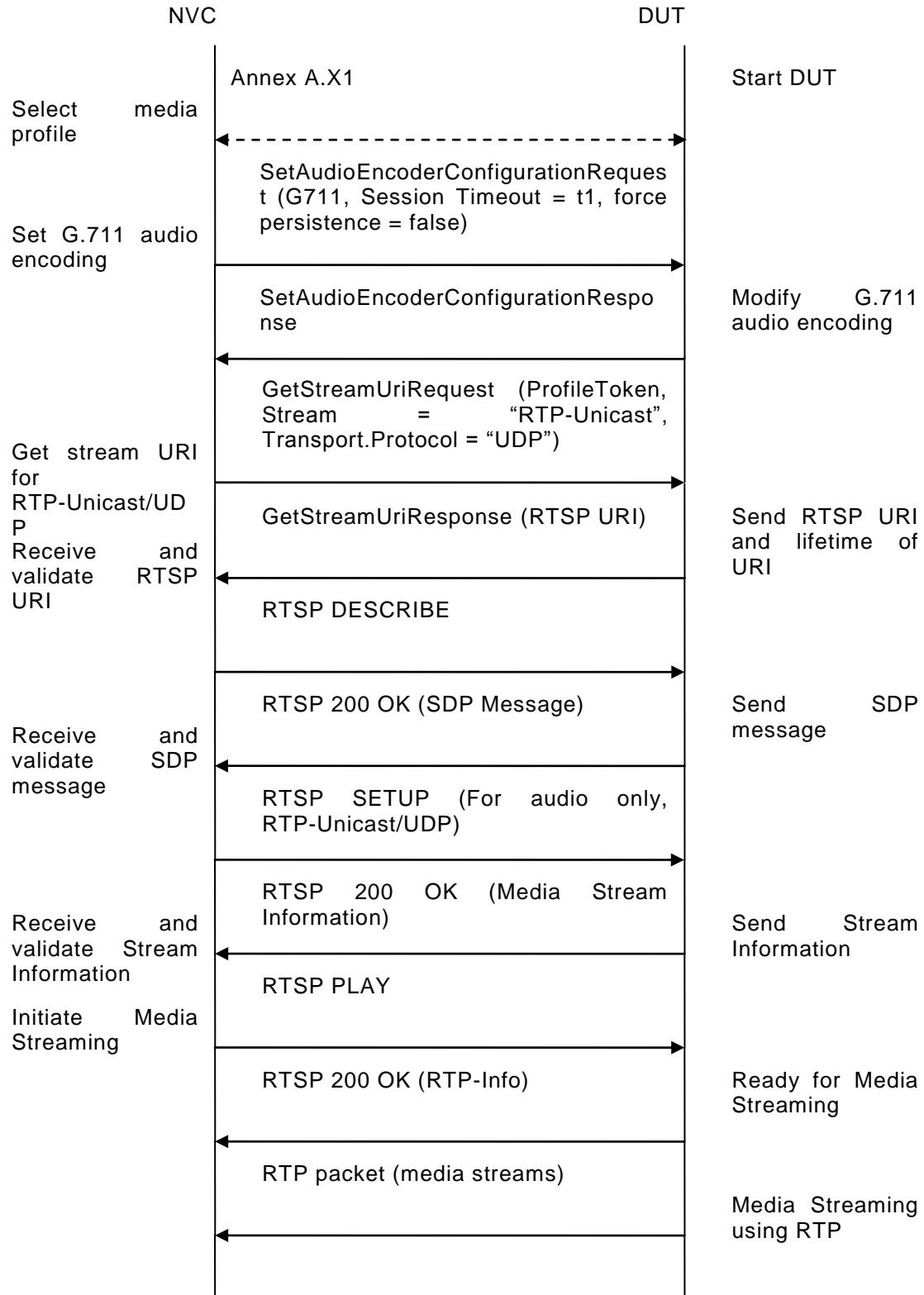
**Requirement Level:** MUST IF SUPPORTED (Audio)

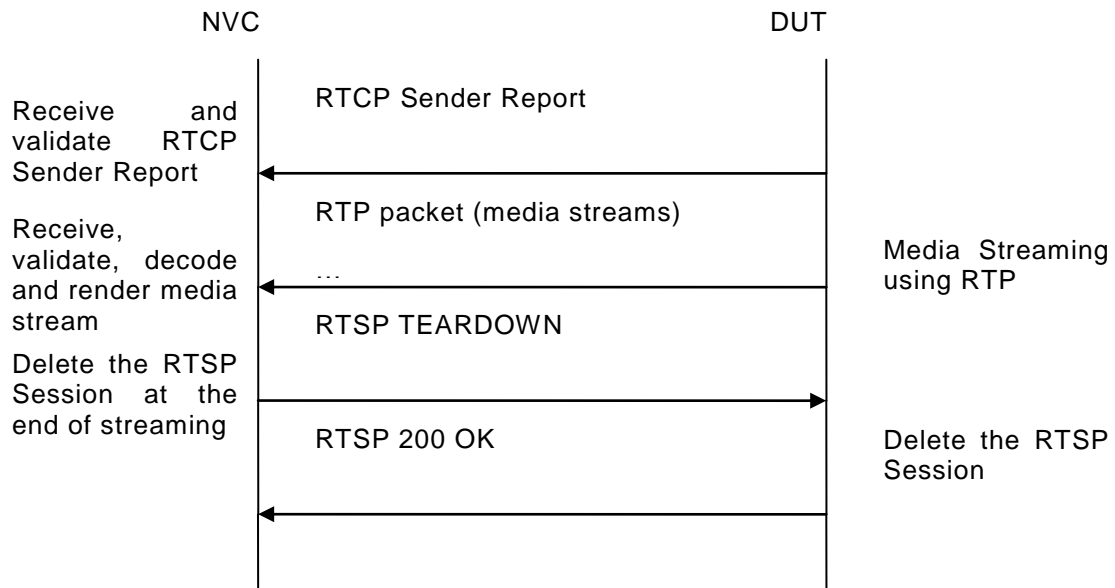
**Test Propose:** To verify G.711 media streaming based on RTP-Unicast/UDP Transport.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with G.711 encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "G711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**UDP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Unicast/UDP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends G.711 RTP media stream to NVC over UDP.
16. The DUT sends RTCP sender report to NVC.

17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send G.711 RTP media streaming to NVC.

The DUT sent not only G.711 RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

## 5.2.2 MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP)

**Test Label:** Real Time Viewing DUT G.711 Media Streaming Using RTP-Unicast/RTSP/HTTP/TCP Transport.

**Test Case ID:** RTSS-2-1-2

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio)

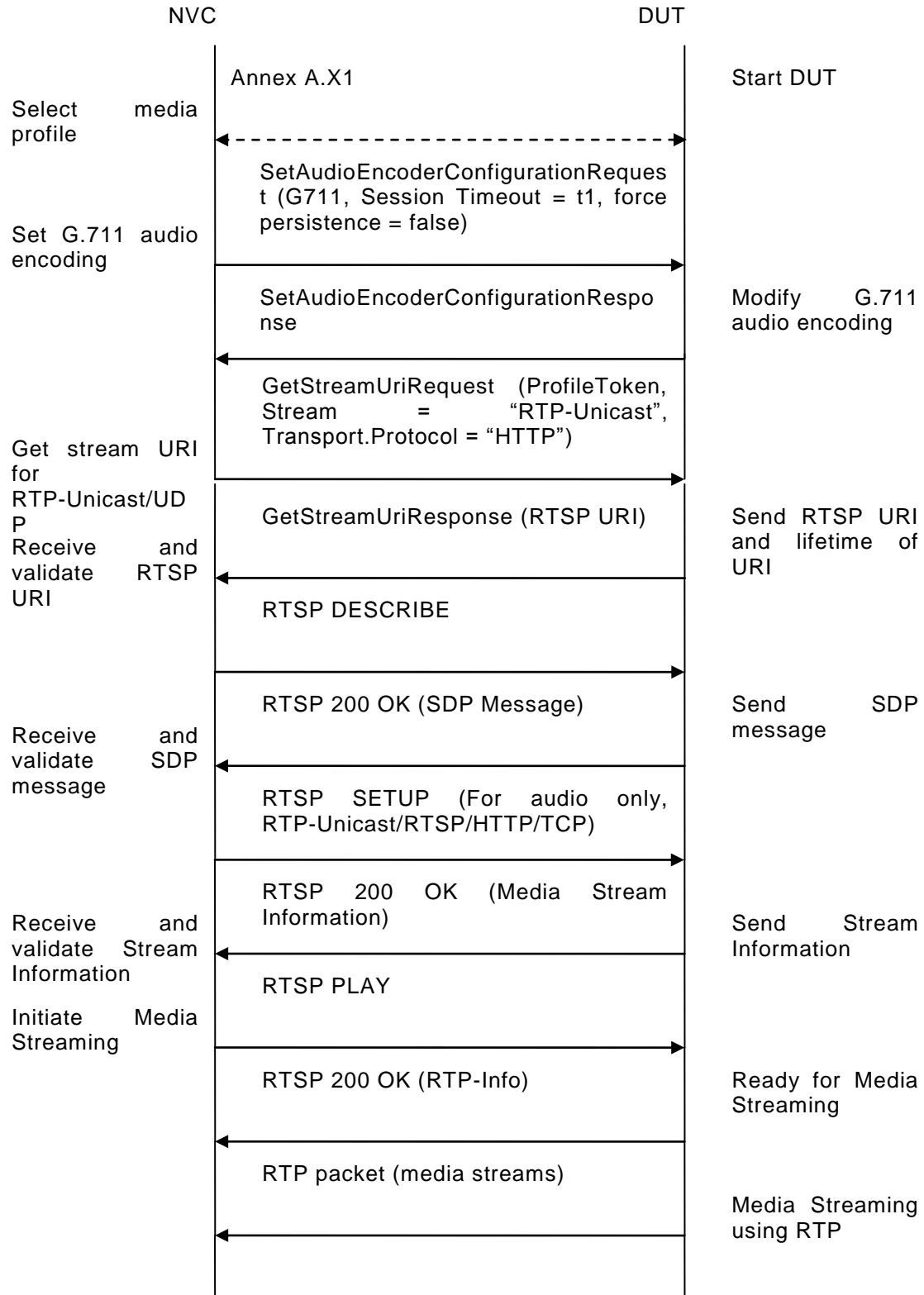
**Test Propose:** To verify G.711 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport.

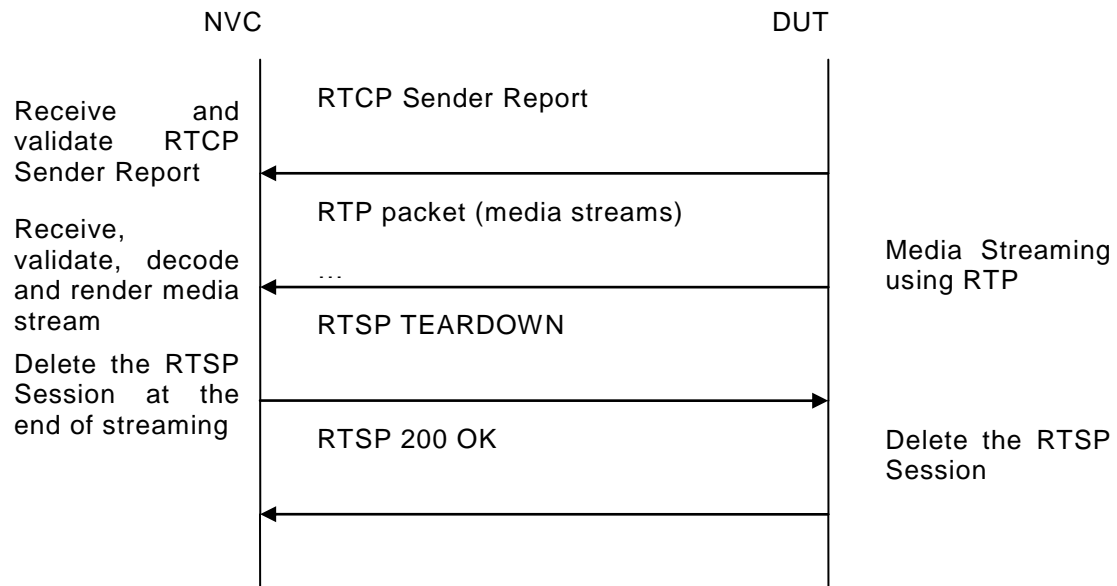
**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**

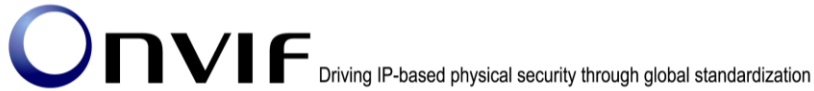






#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with G.711 encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "G711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**HTTP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Unicast/RTSP/HTTP/TCP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends G.711 RTP media stream to NVC over HTTP.
16. The DUT sends RTCP sender report to NVC.



17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send G.711 RTP media streaming to NVC.

The DUT sent not only G.711 RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### **5.2.3 MEDIA STREAMING – G.711 (RTP/RTSP/TCP)**

**Test Label:** Real Time Viewing DUT G.711 Media Streaming Using RTP/RTSP/TCP Transport.

**Test Case ID:** RTSS-2-1-3

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

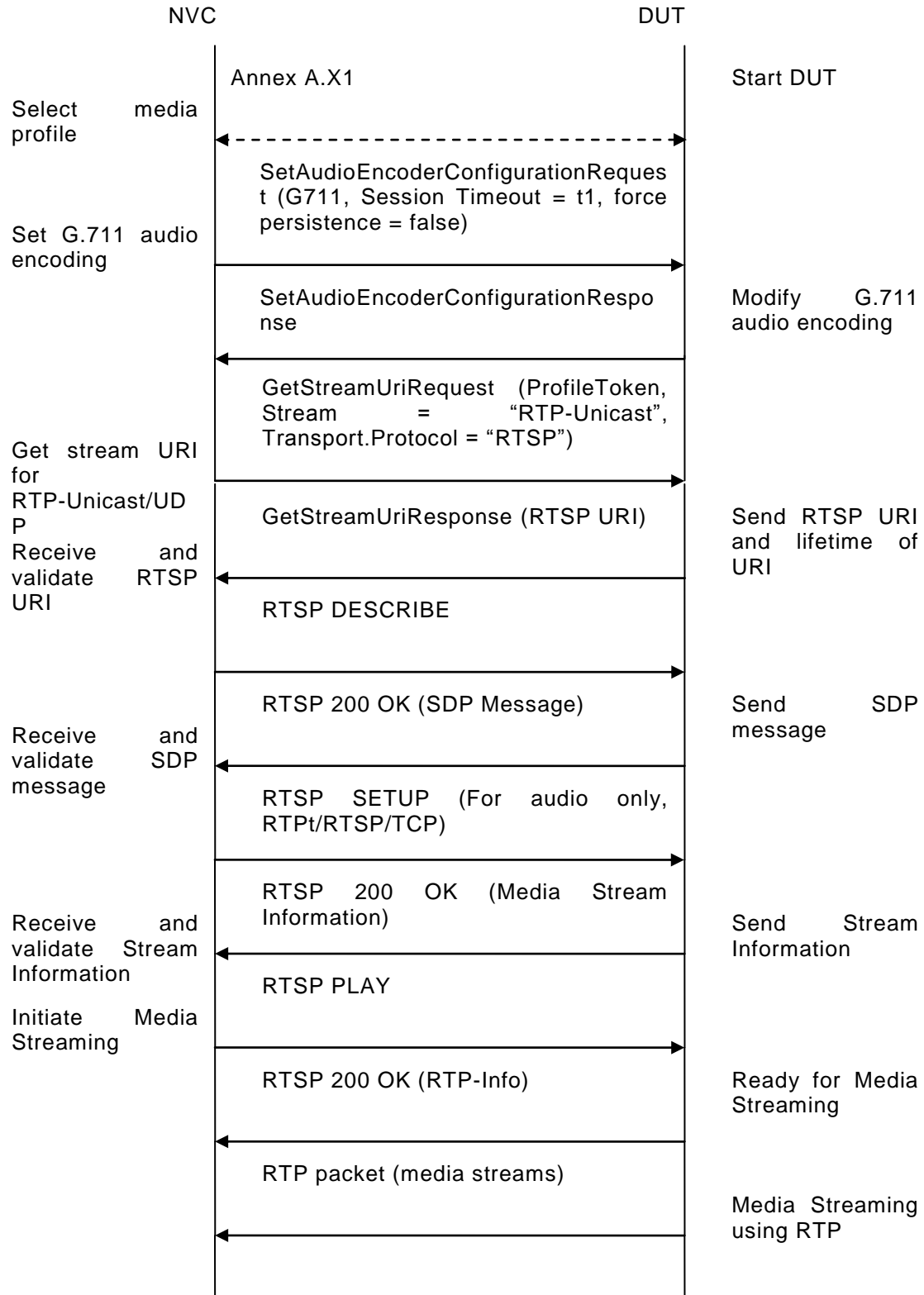
**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (RTP/RTSP/TCP)

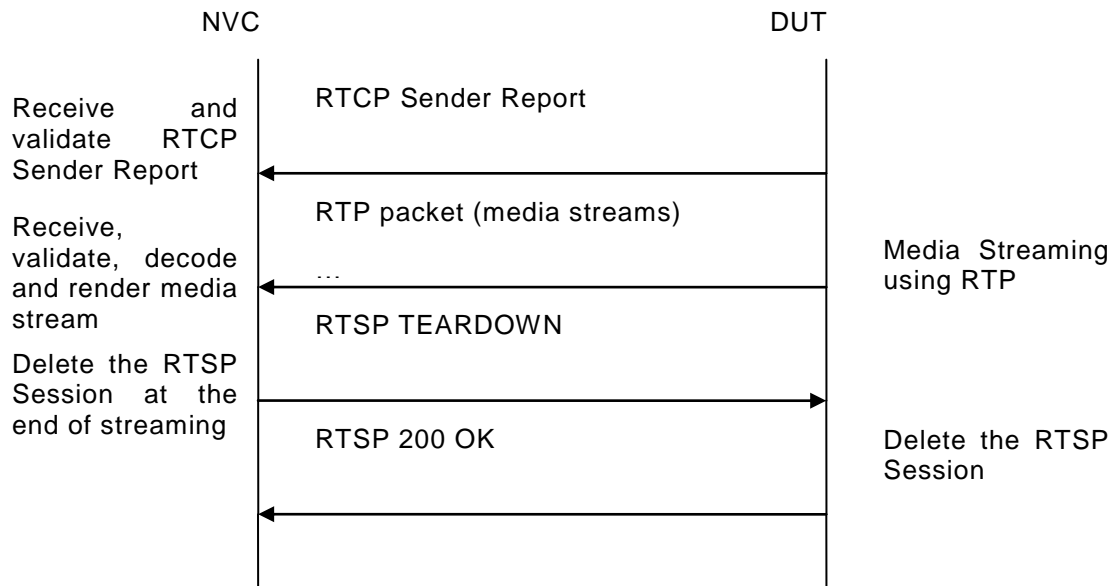
**Test Propose:** To verify G.711 media streaming based on RTP/RTSP/TCP Transport.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

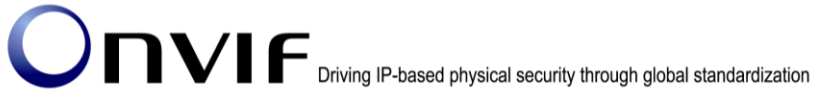
**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with G.711 encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "G711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**RTSP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP/RTSP/TCP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends G.711 RTP media stream to NVC over RTSP.
16. The DUT sends RTCP sender report to NVC.



17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send G.711 RTP media streaming to NVC.

The DUT sent not only G.711 RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

**5.2.4 MEDIA STREAMING – G.726 (RTP-Unicast/UDP)**

**Test Label:** Real Time Viewing DUT G.726 Media Streaming Using RTP-Unicast/UDP Transport.

**Test Case ID:** RTSS-2-1-4

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (G.726)

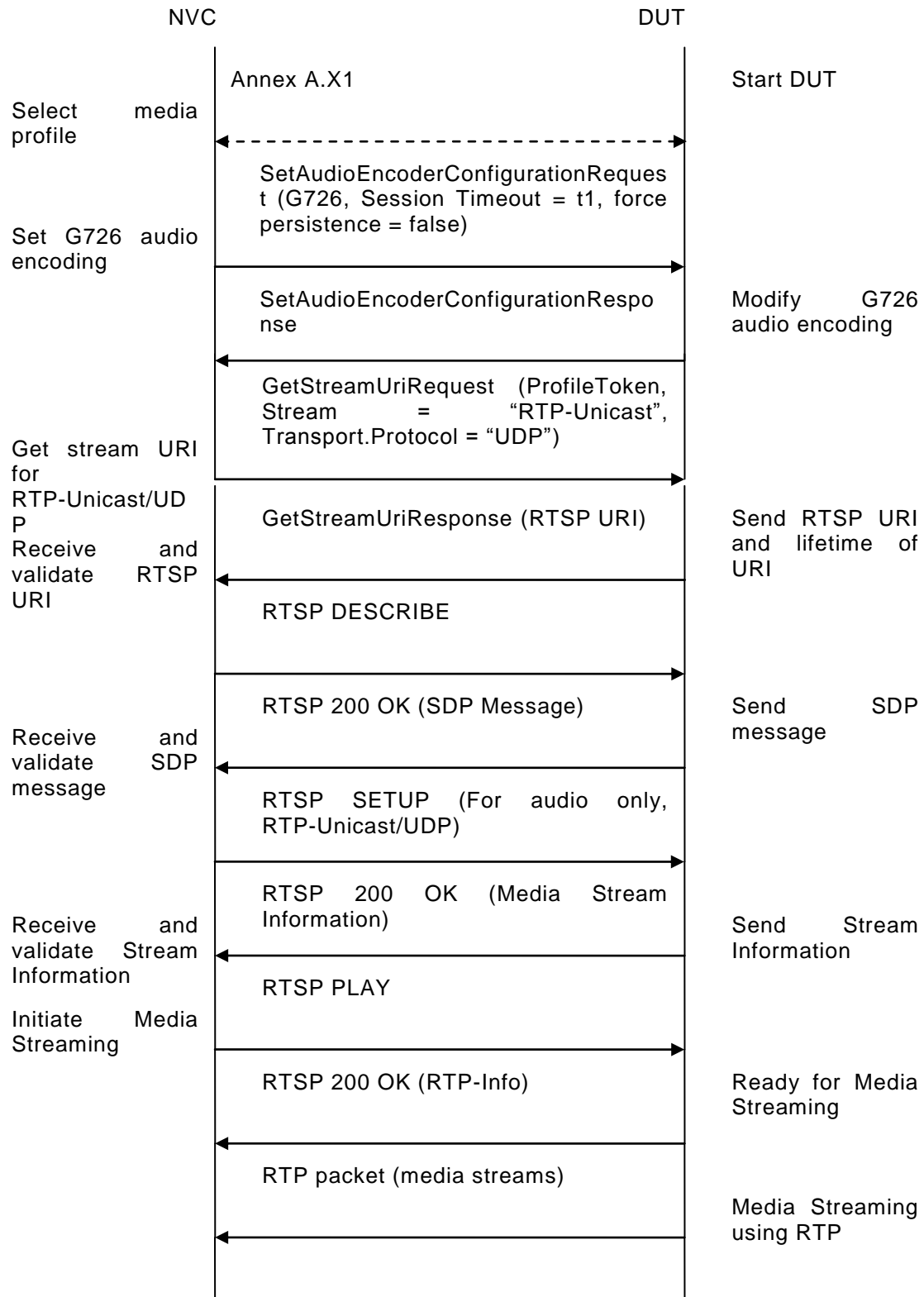
**Test Propose:** To verify G.726 media streaming based on RTP-Unicast/UDP Transport.

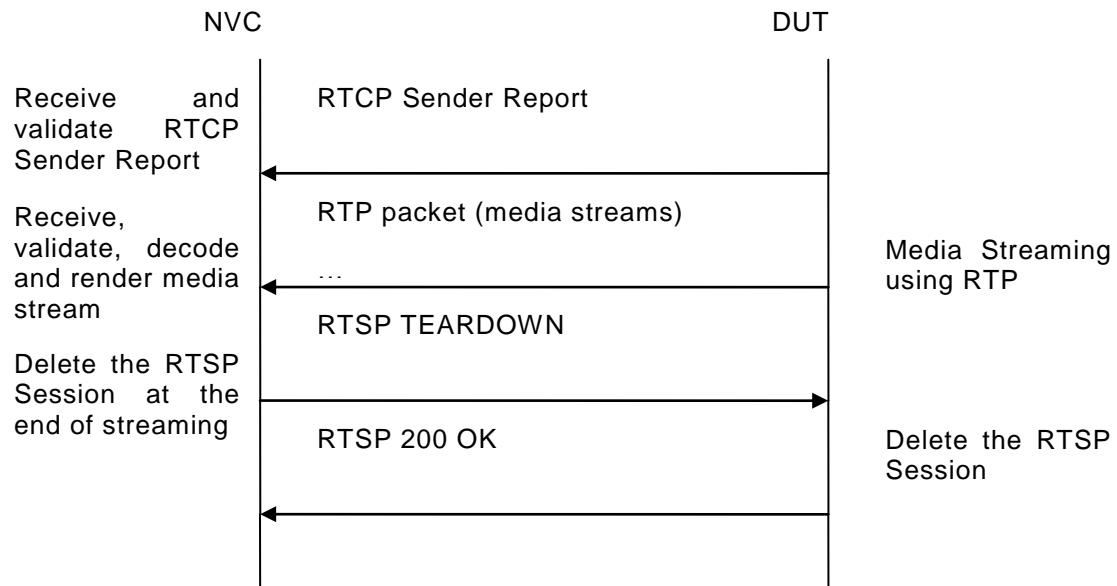
**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**

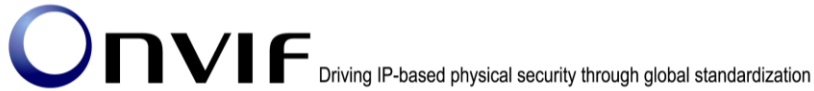






#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with G.726 encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "G726", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**UDP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Unicast/UDP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends G.726 RTP media stream to NVC over UDP.
16. The DUT sends RTCP sender report to NVC.



17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send G.726 RTP media streaming to NVC.

The DUT sent not only G.726 RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

## **5.2.5 MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT G.726 Media Streaming Using RTP-Unicast/RTSP/HTTP/TCP Transport.

**Test Case ID:** RTSS-2-1-5

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

**Command Under Test:** None

**WSDL Reference:** None

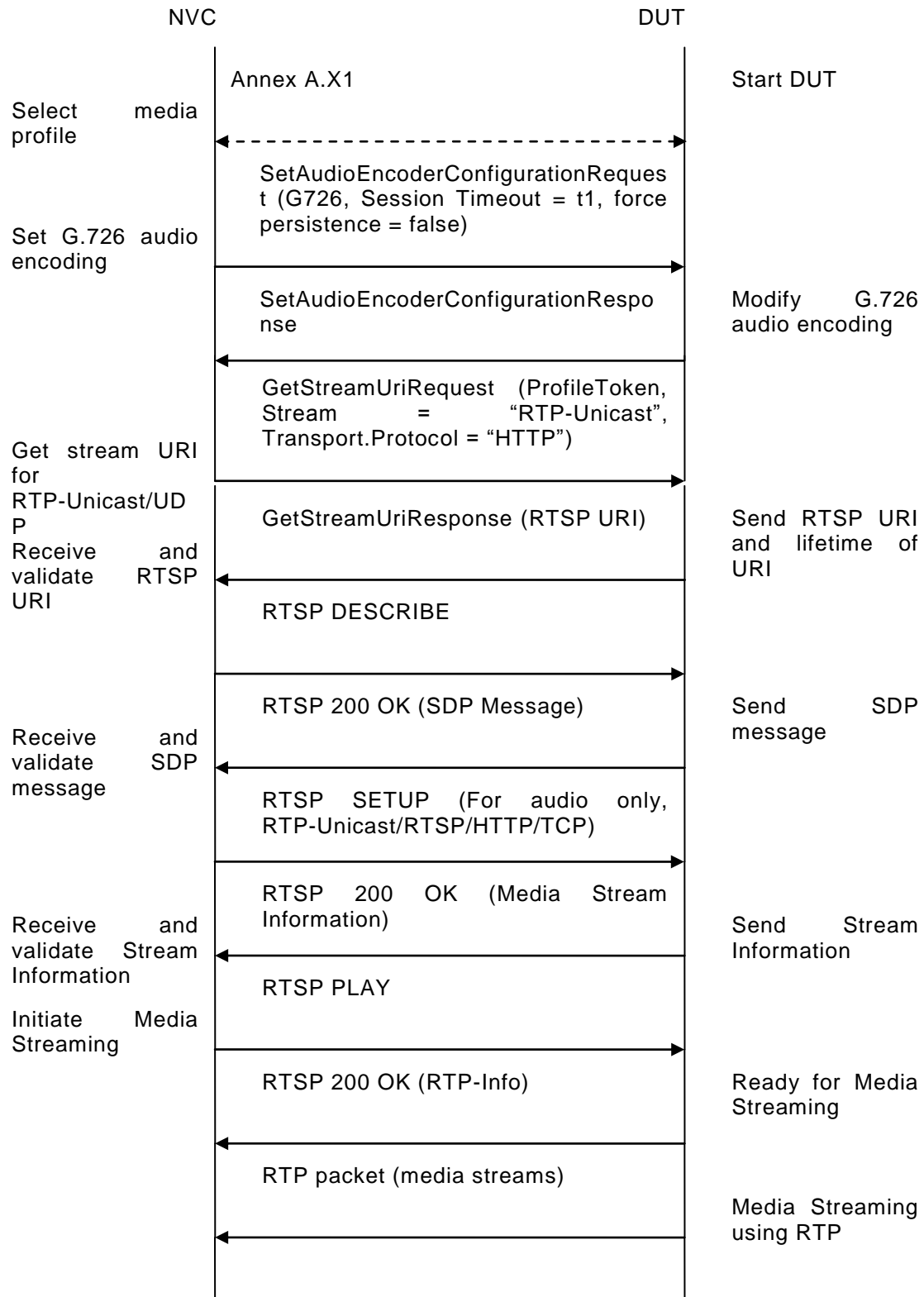
**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (G.726)

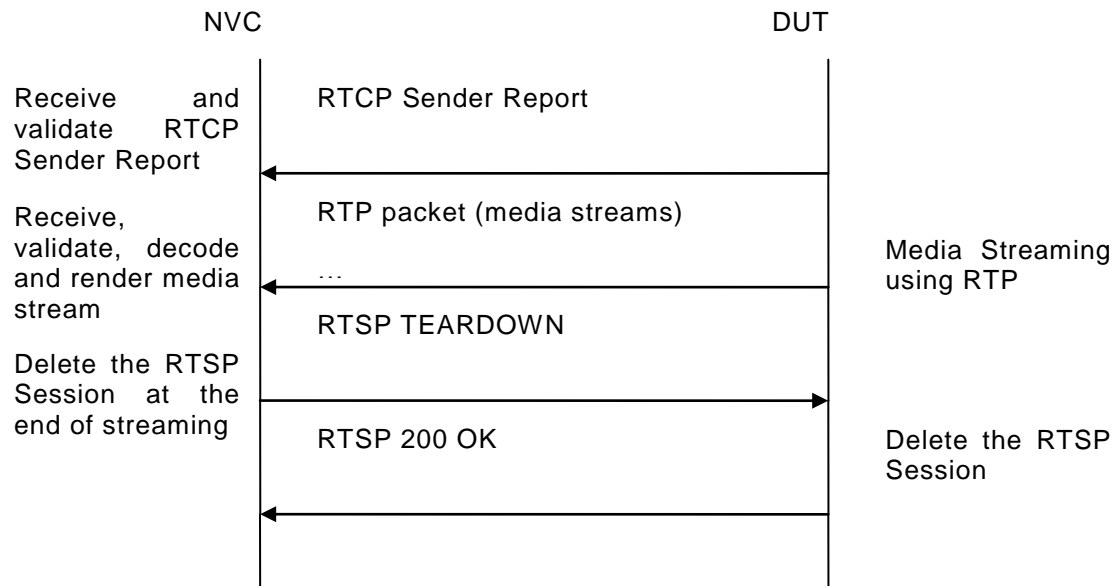
**Test Propose:** To verify G.726 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

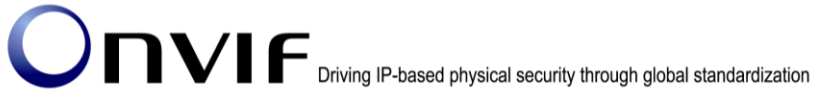
**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with G.726 encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "G726", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**HTTP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Unicast/RTSP/HTTP/TCP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends G.726 RTP media stream to NVC over HTTP.
16. The DUT sends RTCP sender report to NVC.



17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send G.726 RTP media streaming to NVC.

The DUT sent not only G.726 RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

## **5.2.6 MEDIA STREAMING – G.726 (RTP/RTSP/TCP)**

**Test Label:** Real Time Viewing DUT G.726 Media Streaming Using RTP/RTSP/TCP Transport.

**Test Case ID:** RTSS-2-1-6

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (G.726 and RTP/RTSP/TCP)

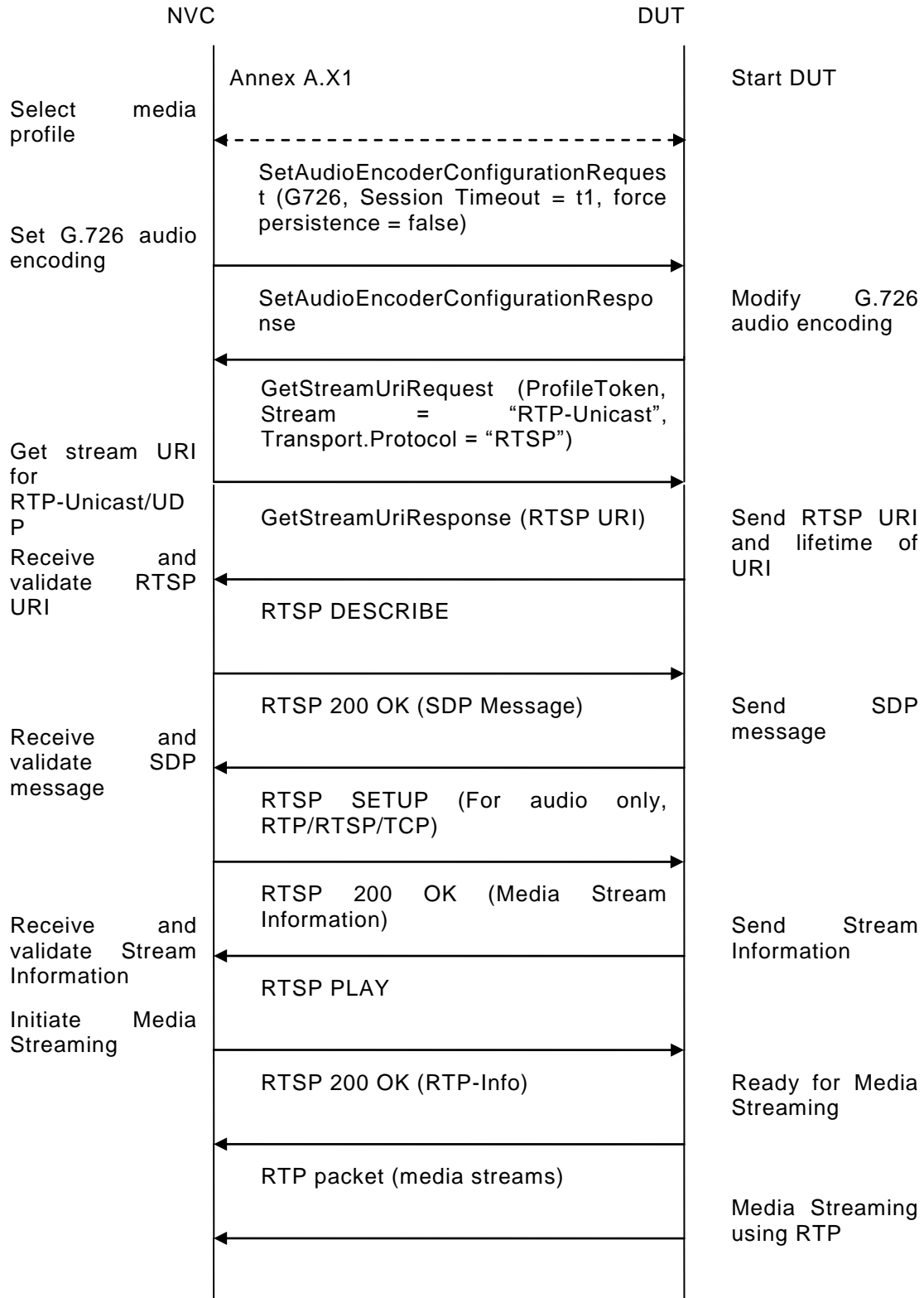
**Test Propose:** To verify G.726 media streaming based on RTP/RTSP/TCP Transport.

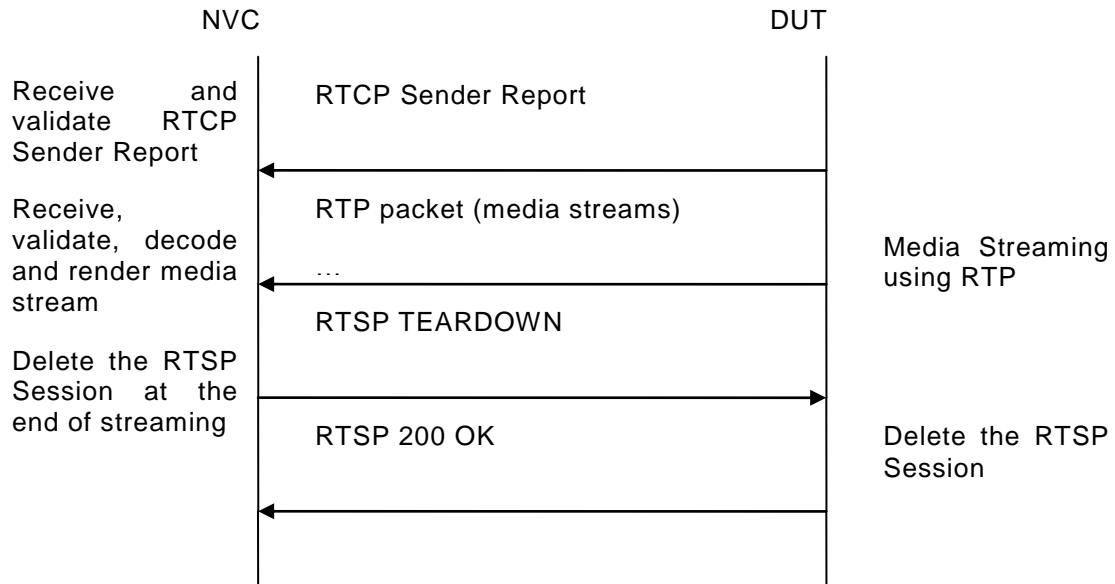
**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

Test Sequence:







#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with G.726 encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "G726", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "RTP-Unicast", **Transport.Protocol** = "RTSP") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP/RTSP/TCP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends G.726 RTP media stream to NVC over RTSP.
16. The DUT sends RTCP sender report to NVC.

17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send G.726 RTP media streaming to NVC.

The DUT sent not only G.726 RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

#### 5.2.7 MEDIA STREAMING – AAC (RTP-Unicast/UDP)

**Test Label:** Real Time Viewing DUT AAC Media Streaming Using RTP-Unicast/UDP Transport.

**Test Case ID:** RTSS-2-1-7

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

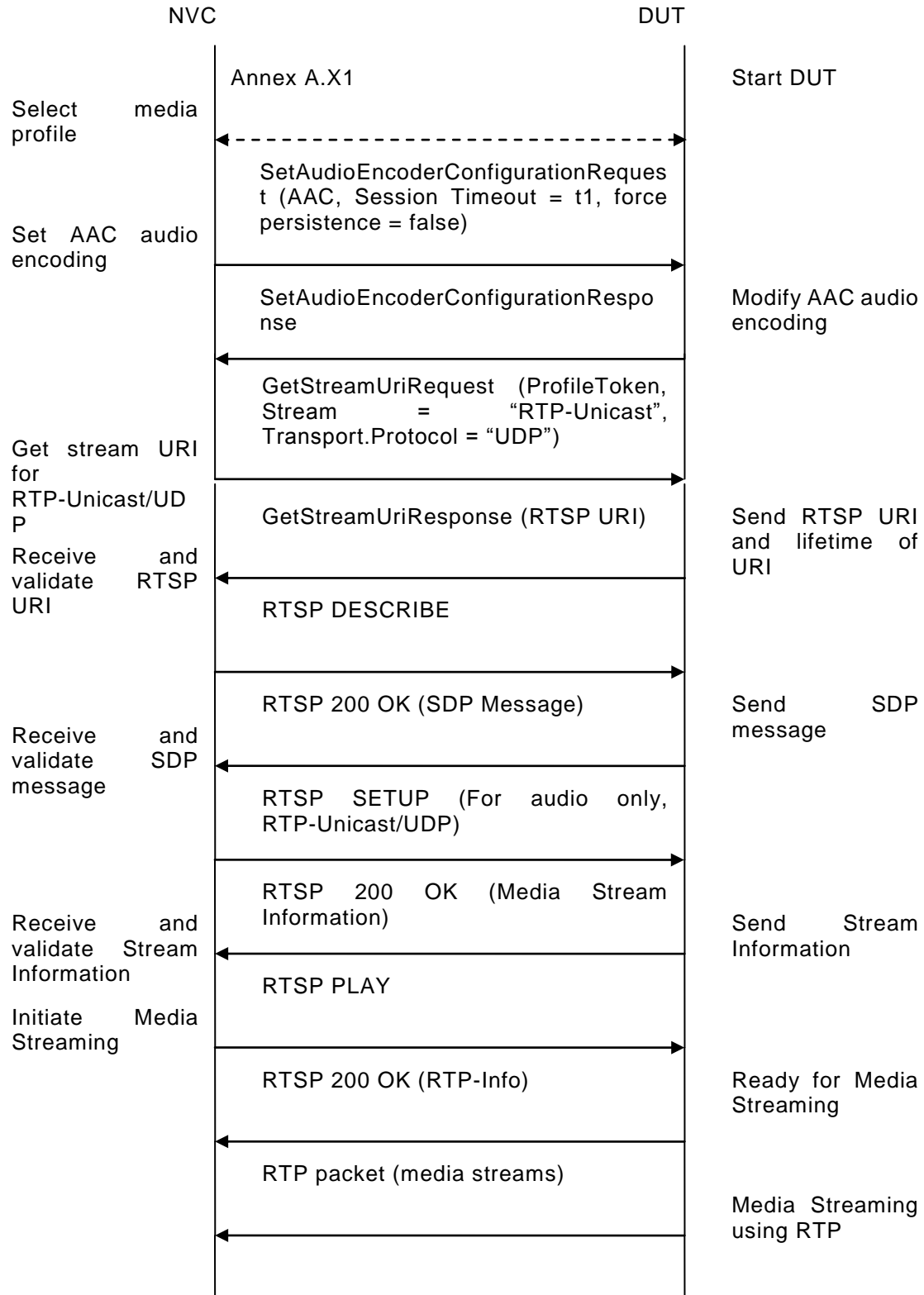
**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (AAC)

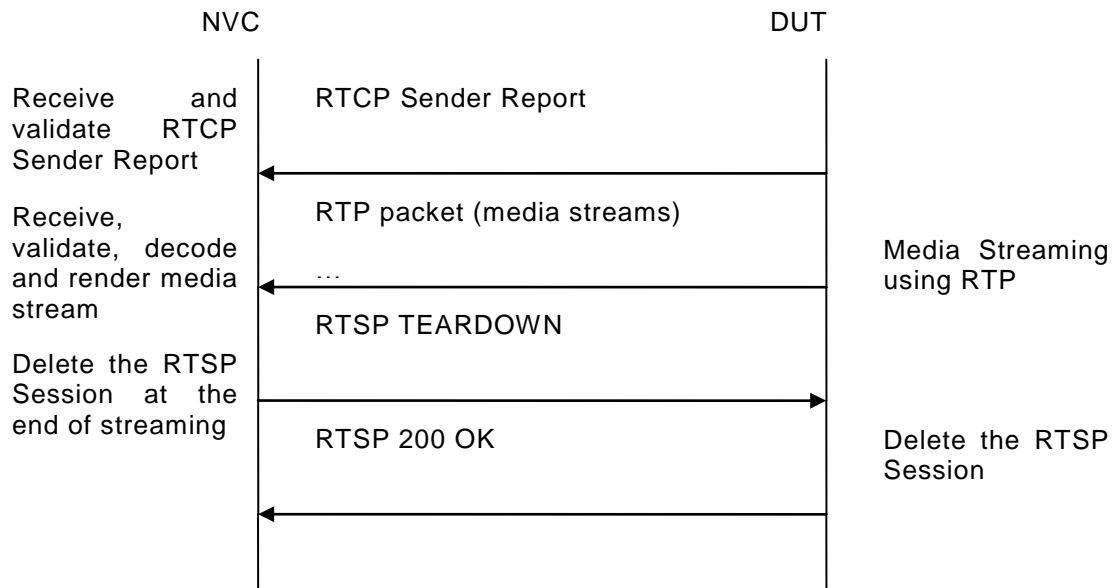
**Test Propose:** To verify AAC media streaming based on RTP-Unicast/UDP Transport.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with AAC encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "AAC", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**UDP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Unicast/UDP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends AAC RTP media stream to NVC over UDP.
16. The DUT sends RTCP sender report to NVC.



17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send AAC RTP media streaming to NVC.

The DUT sent not only AAC RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

## **5.2.8 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT AAC Media Streaming Using RTP-Unicast/RTSP/HTTP/TCP Transport.

**Test Case ID:** RTSS-2-1-8

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (AAC)

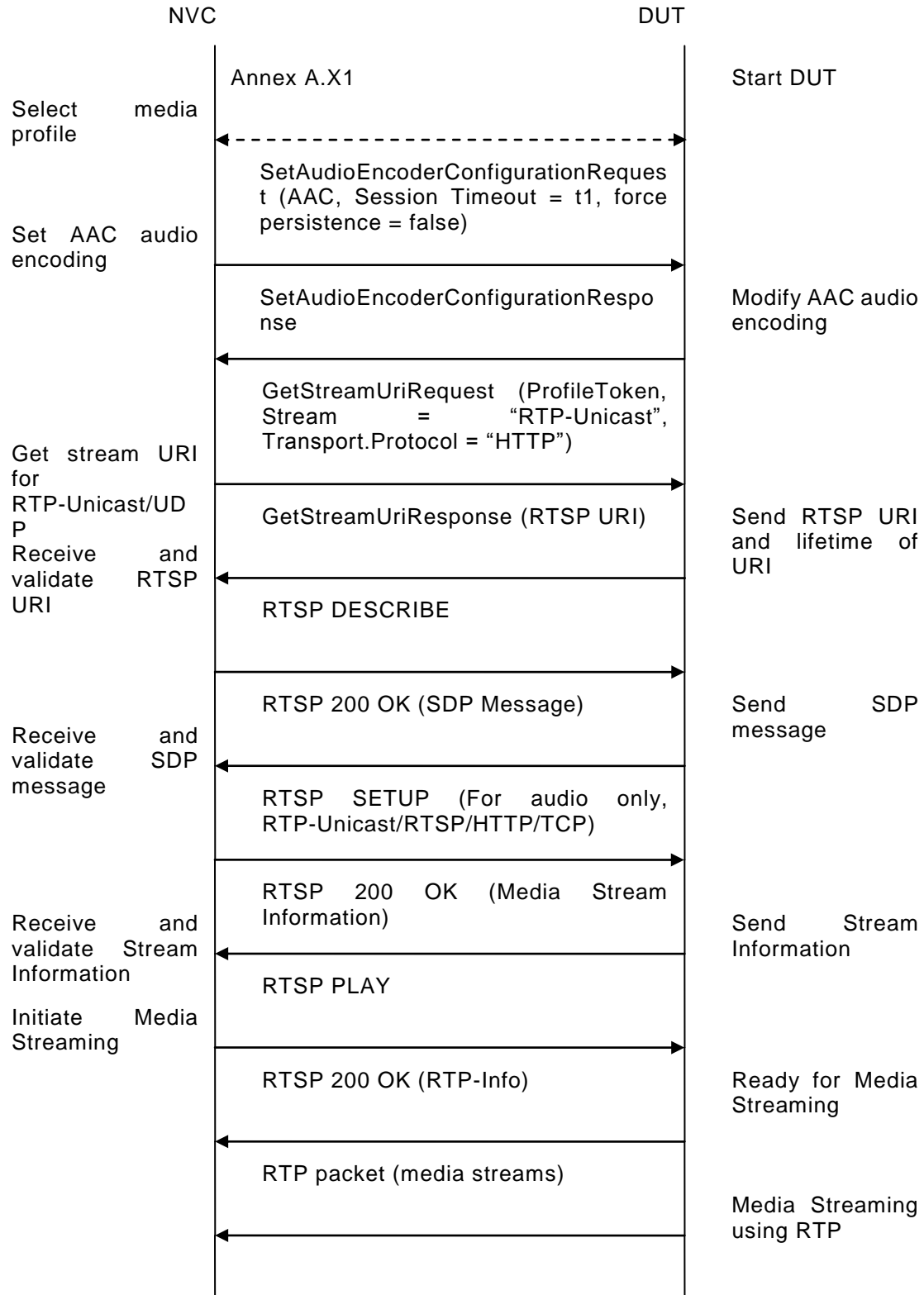
**Test Propose:** To verify AAC media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport.

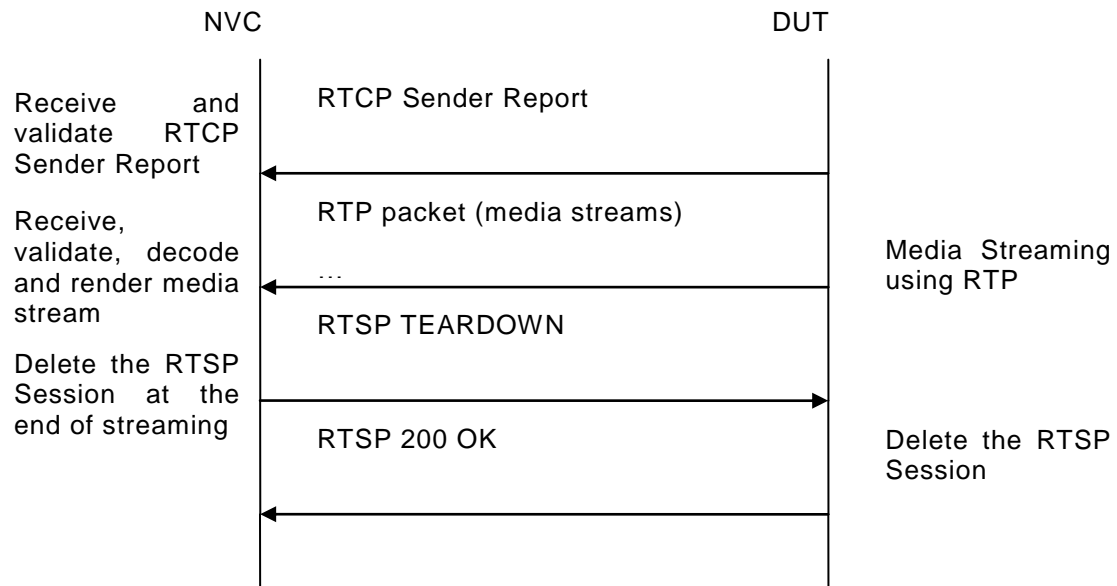
**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with AAC encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "AAC", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "**RTP-Unicast**", **Transport.Protocol** = "**HTTP**") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP-Unicast/RTSP/HTTP/TCP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends AAC RTP media stream to NVC over HTTP.
16. The DUT sends RTCP sender report to NVC.

17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send AAC RTP media streaming to NVC.

The DUT sent not only AAC RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.2.9 MEDIA STREAMING – AAC (RTP/RTSP/TCP)

**Test Label:** Real Time Viewing DUT AAC Media Streaming Using RTP/RTSP/TCP Transport.

**Test Case ID:** RTSS-2-1-9

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, TCP, Stream control, RTSP

**Command Under Test:** None

**WSDL Reference:** None

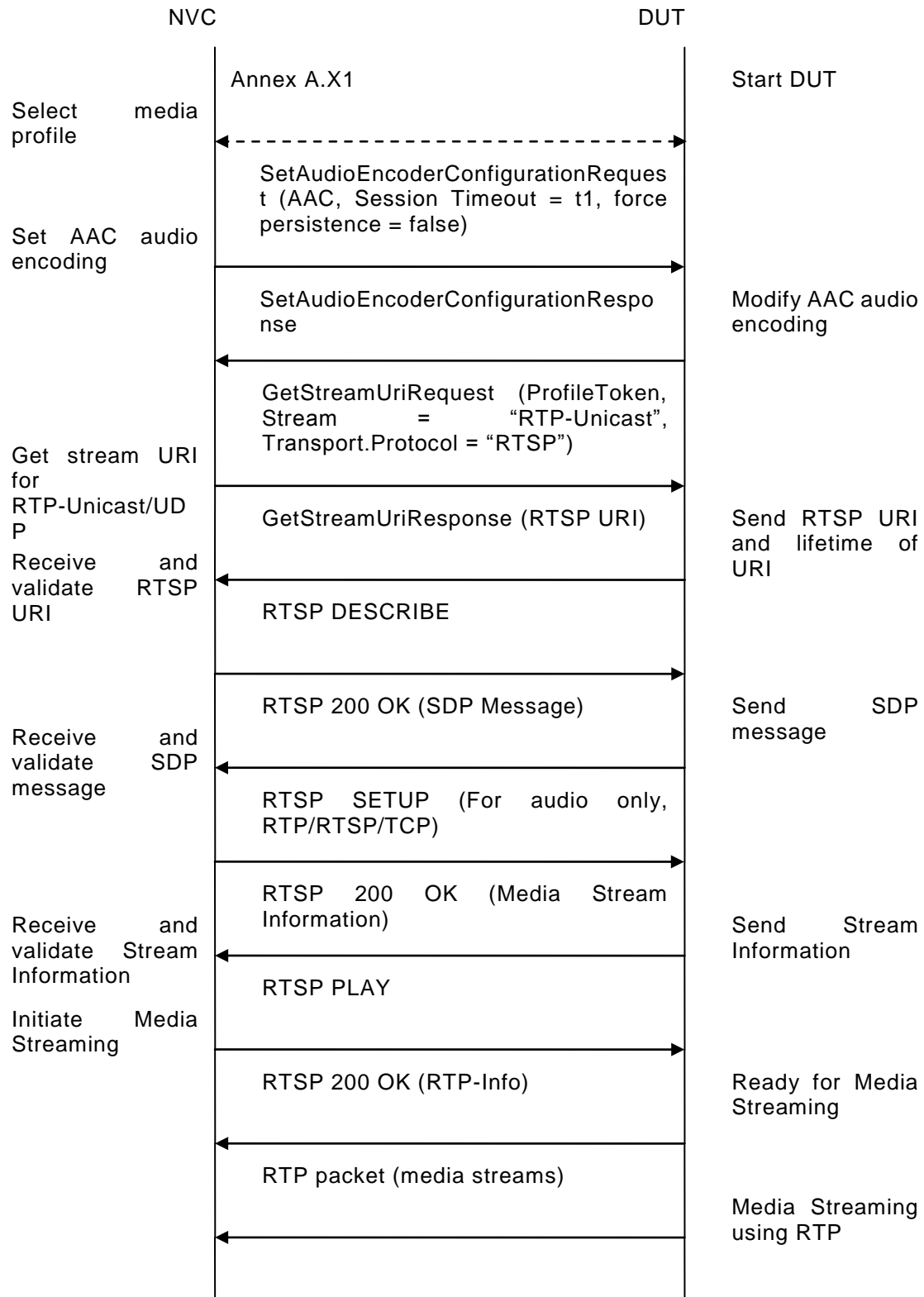
**Requirement Level:** MUST IF SUPPORTED (Audio) AND IMPLEMENTED (AAC and RTP/RTSP/TCP)

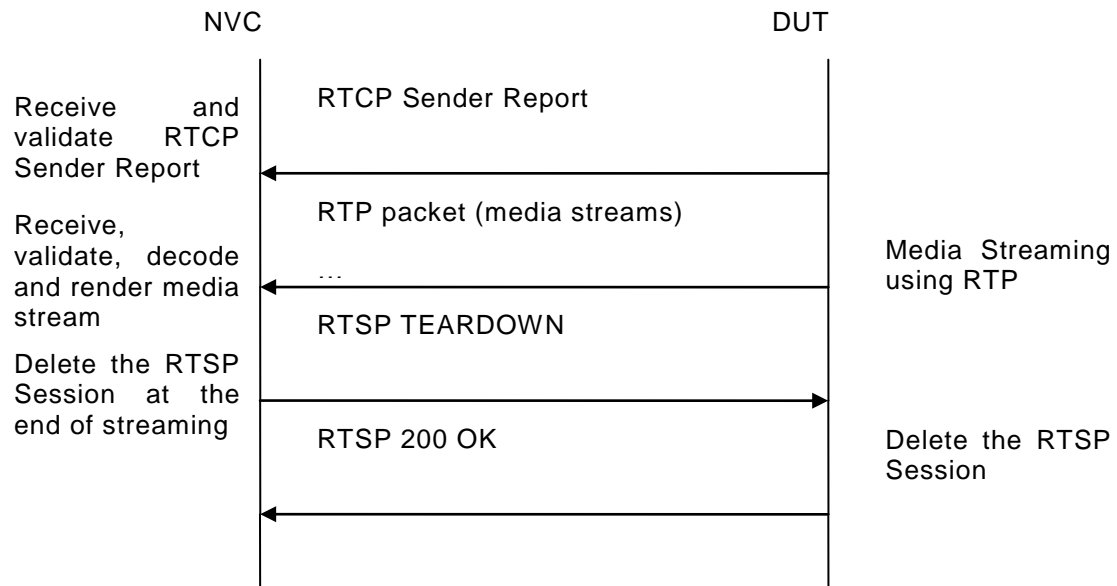
**Test Propose:** To verify AAC media streaming based on RTP/RTSP/TCP Transport.

**Pre-Requisite:** Media is supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. NVC gets the Media Service entry point by GetCapabilities command.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start a DUT.
3. NVC selects a media profile with AAC encoding support.
4. NVC invokes SetAudioEncoderConfigurationRequest message (Encoding = "AAC", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.
5. Verify the SetAudioEncoderConfigurationResponse message from the DUT.
6. NVC invokes GetStreamUriRequest message (**ProfileToken**, **Stream** = "RTP-Unicast", **Transport.Protocol** = "RTSP") to retrieve media stream URI for the selected media profile.
7. The DUT sends the GetStreamUriResponse message with RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout.
8. NVC verifies the RTSP media stream URI provided by the DUT.
9. NVC invokes RTSP DESCRIBE request.
10. The DUT sends 200 OK message and SDP information.
11. NVC invokes RTSP SETUP request with transport parameter **RTP/RTSP/TCP**.
12. The DUT sends 200 OK message and the media stream information.
13. NVC invokes RTSP PLAY request.
14. The DUT sends 200 OK message and starts media streaming.
15. The DUT sends AAC RTP media stream to NVC over RTSP.
16. The DUT sends RTCP sender report to NVC.

17. NVC validates the received RTP and RTCP packets, decodes and renders them.
18. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. The DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:**

**PASS –**

DUT passes all assertions.

**FAIL –**

The DUT did not have valid media profile.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

The DUT did not send valid GetStreamUriResponse message.

The DUT did not send valid RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout in GetStreamUriResponse message.

The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

The DUT did not send valid RTP header in one or more media streams.

The DUT did not send RTCP sender report correctly.

The DUT did not send AAC RTP media streaming to NVC.

The DUT sent not only AAC RTP media streaming to NVC.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

## 5.3 Audio & Video Streaming

### 5.3.1 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/ UDP)

**Test Label:** Real Time Viewing DUT JPEG/G.711 Audio&Video streaming using RTP-Unicast/UDP transport.

**Test Case ID:** RTSS-3-1-1

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio)

**Test Purpose:** To verify JPEG/G.711 Audio&Video streaming based on RTP/UDP Unicast Transport.

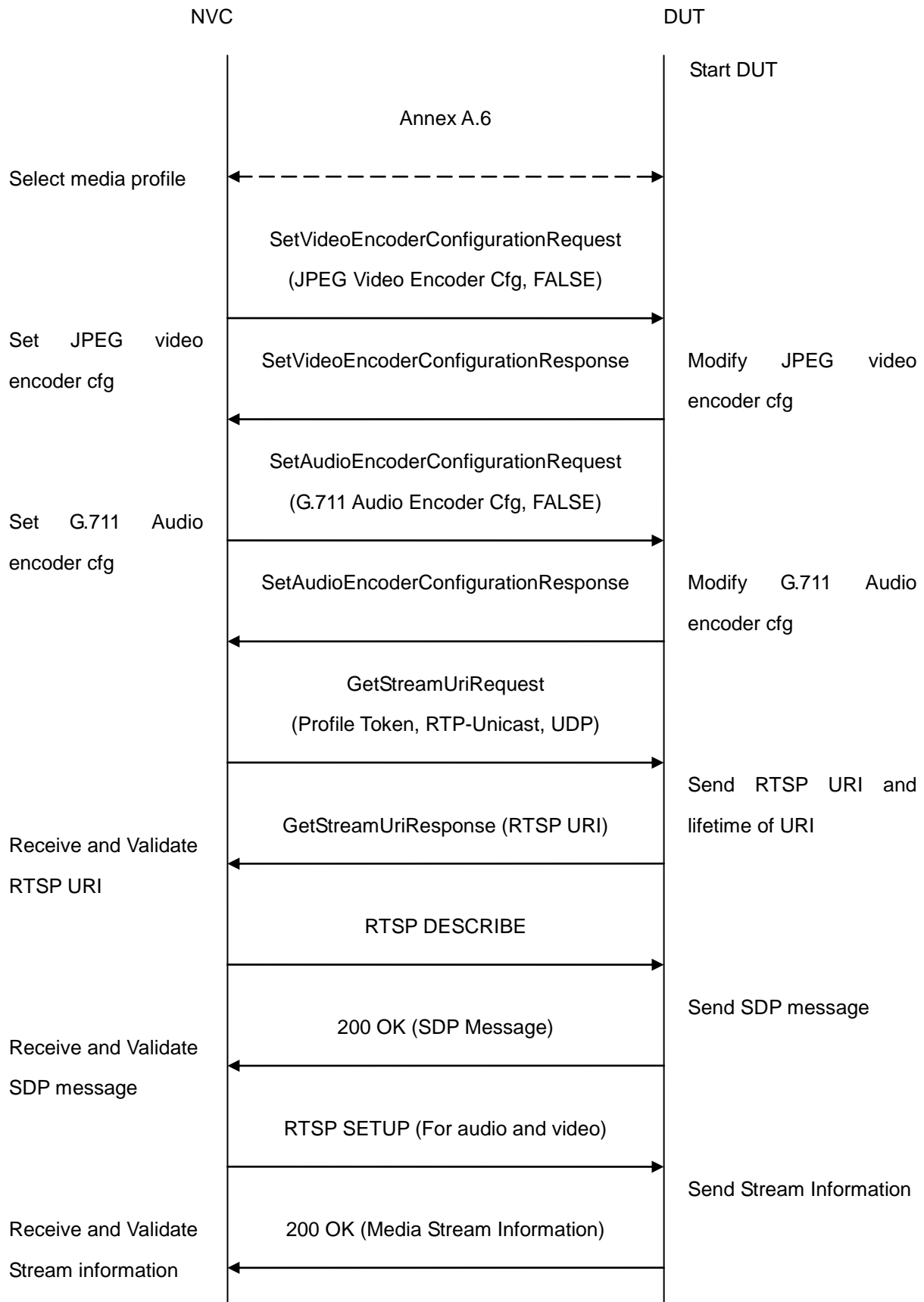
**Pre-Requisite:** Audio is supported by DUT

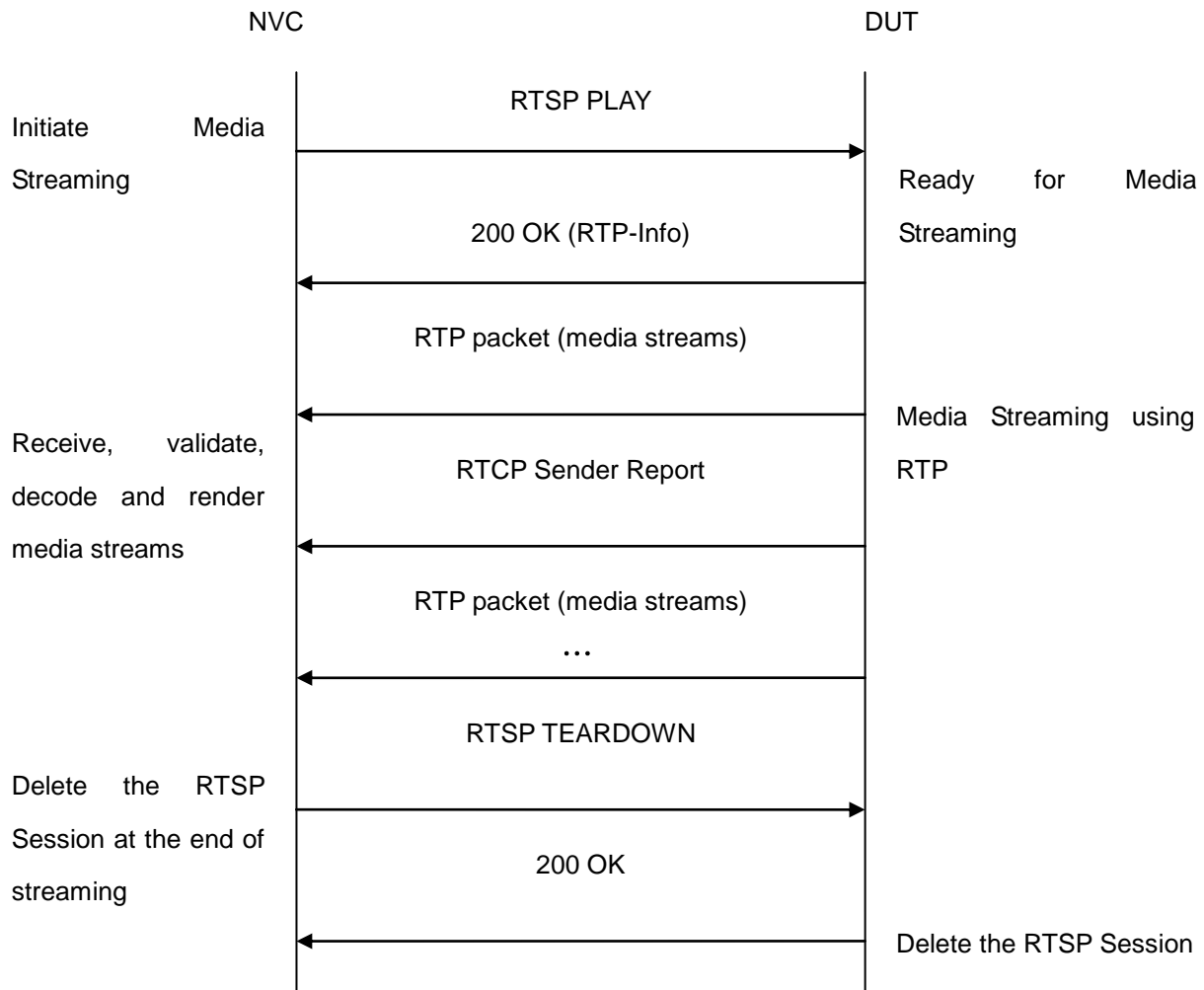
A media profile with JPEG video encoder configuration and G.711 audio encoder configuration

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and G.711 audio encoding support by following the procedure mentioned in Annex A.6.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "G711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.

8. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the RTSP media stream URI provided by the DUT.
11. NVC invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP** for both audio and video streams separately.
14. DUT sends 200 OK message and the media stream information.
15. NVC invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT sends Audio/Video RTP media stream to NVC over UDP.
18. DUT sends Audio/Video RTCP sender report to NVC.
19. DUT validates the received RTP and RTCP packets, decodes and renders them.
20. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
21. DUT sends 200 OK Response and terminates the RTSP Session.

#### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### **5.3.2 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT JPEG/G.711 Audio&Video streaming using HTTP transport.

**Test Case ID:** RTSS-3-1-2

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio)

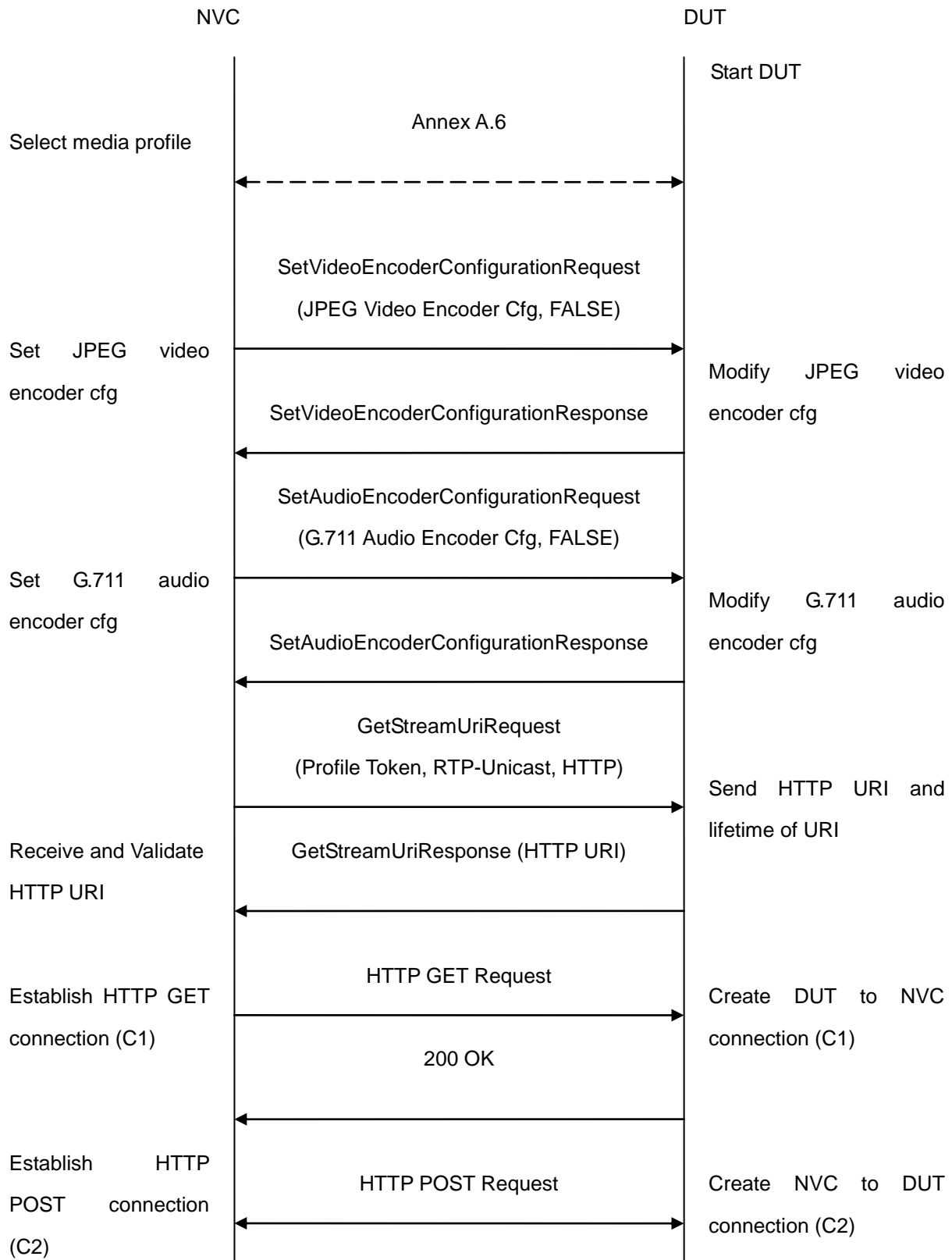
**Test Purpose:** To verify JPEG/G.711 Audio&Video streaming based on HTTP Transport.

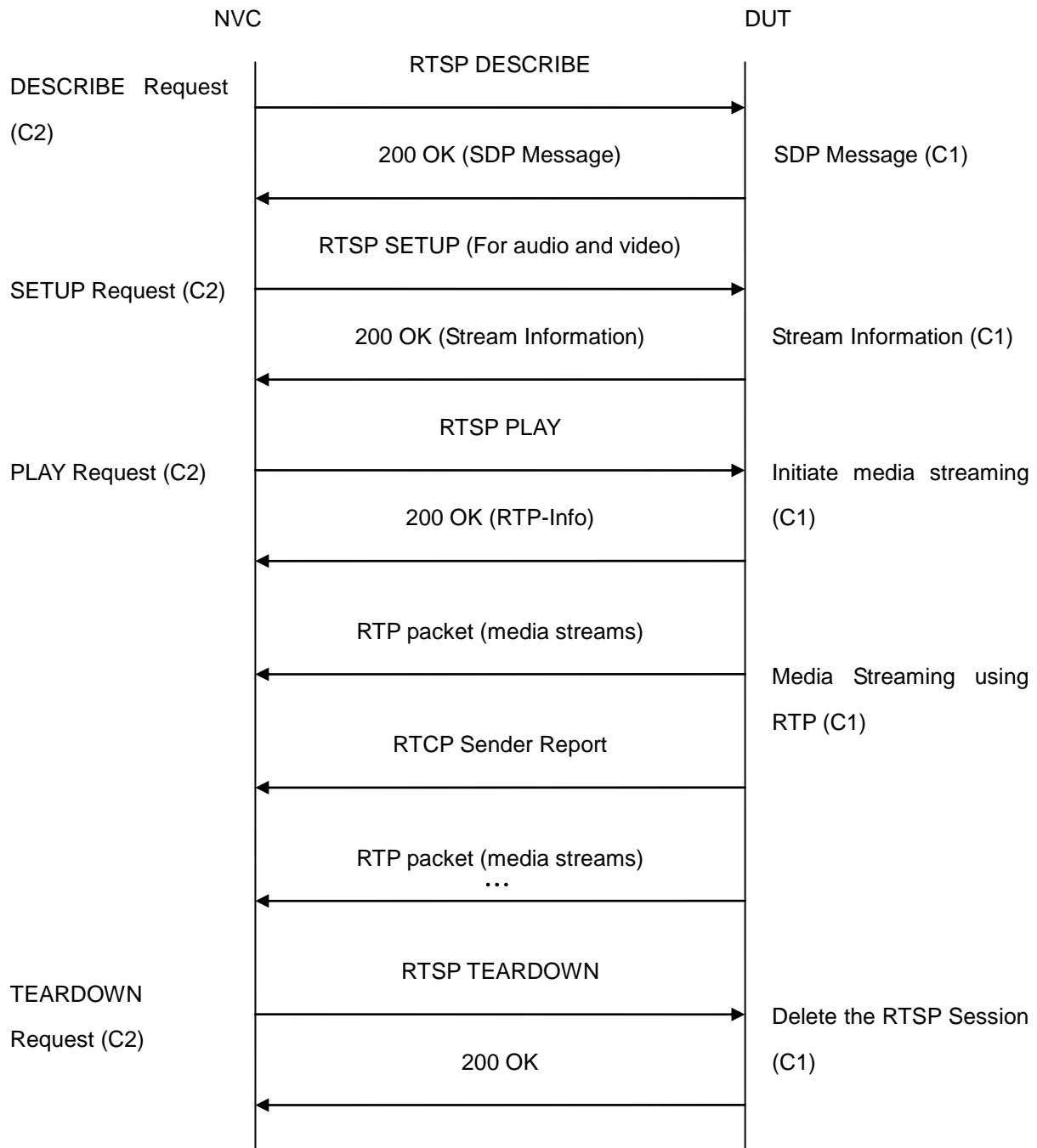
**Pre-Requisite:** Audio is supported by DUT.

A media profile with JPEG video encoder configuration and G.711 audio encoder configuration

**Test Configuration:** NVC and DUT

**Test Sequence:**





**Test Procedure:**

1. Start an NVC.
2. Start an NVC.
3. Start an DUT.
4. NVC selects a media profile with both JPEG video encoding support and G.711 audio encoding support by following the procedure mentioned in Annex A.6.

5. NVC invokes SetVideoEncoderConfigurationRequest (Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
6. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
7. NVC invokes SetAudioEncoderConfigurationRequest (Encoding = "G711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
8. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
9. NVC invokes GetStreamUriRequest message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.
10. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
11. NVC verifies the HTTP media stream URI provided by the DUT.
12. NVC invokes HTTP GET Request on DUT and establishes DUT to NVC connection for RTP data transfer.
13. NVC invokes HTTP POST Request and establishes NVC to DUT connection for RTSP control requests.
14. NVC invokes RTSP DESCRIBE request on HTTP POST connection.
15. DUT sends 200 OK message and SDP information on HTTP GET connection.
16. NVC invokes RTSP SETUP requests on HTTP POST connection with transport parameter as 'RTP/TCP' along with 'interleaved' parameter for both audio and video streams separately.
17. DUT sends 200 OK message and the media stream information on HTTP GET connection.
18. NVC invokes RTSP PLAY request on HTTP POST connection.
19. DUT sends 200 OK message and starts media streaming on HTTP GET connection.
20. DUT transfers Audio/Video RTP media stream to NVC on HTTP GET connection.
21. DUT sends Audio/Video RTCP sender report to NVC on HTTP GET connection.
22. DUT validates the received RTP and RTCP packets, decodes and renders them.
23. NVC invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.
24. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.3.3 MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP)

**Test Label:** Real Time Viewing DUT JPEG/G.711 Audio&Video streaming using RTP/RTSP/TCP transport.

**Test Case ID:** RTSS-3-1-3

**ONVIF Core Specification Coverage:** RTP/RTSP/TCP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (RTP/RTSP/TCP)

**Test Purpose:** To verify JPEG/G.711 Audio&Video streaming based on RTP/RTSP/TCP using RTSP tunnel.

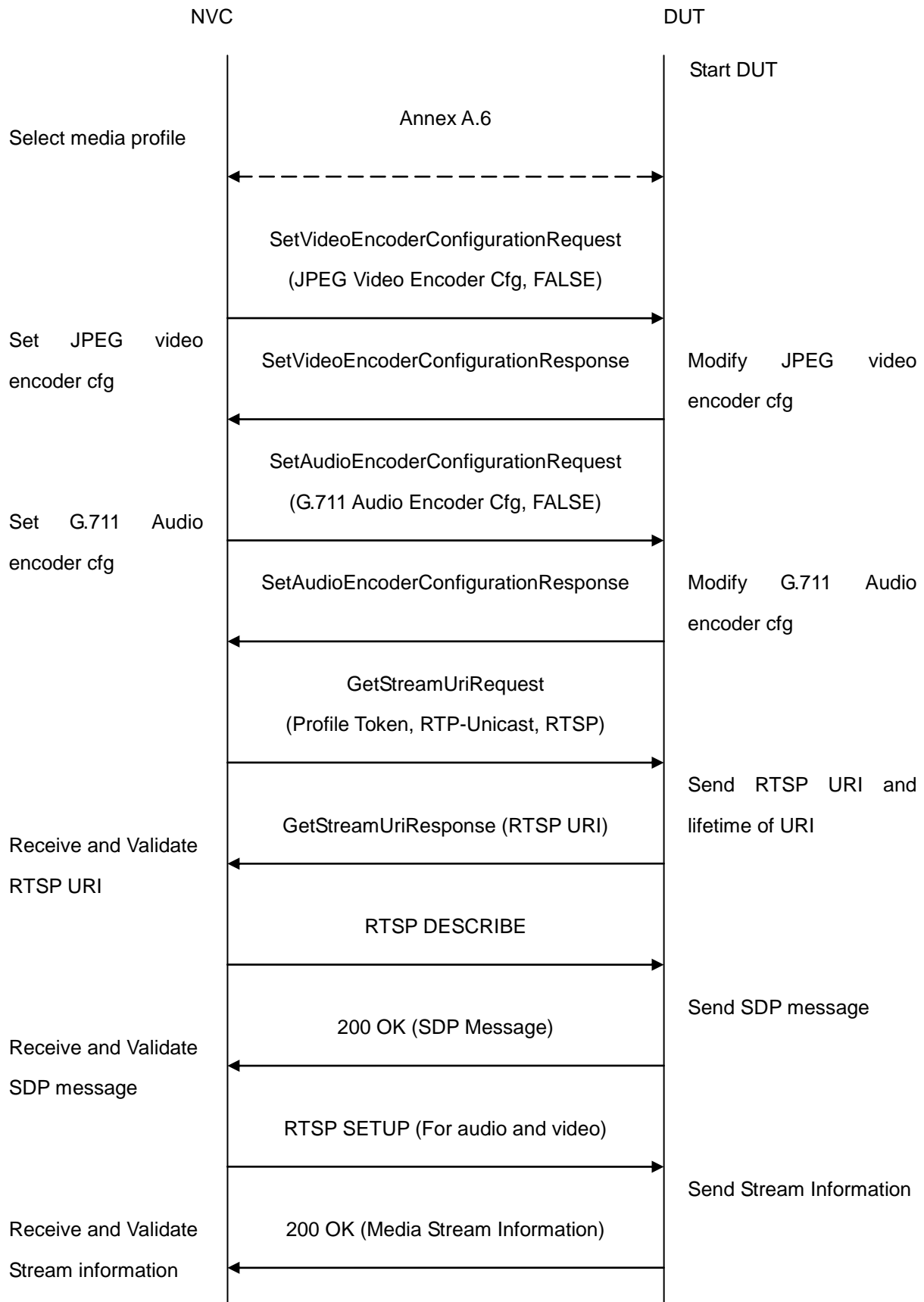
**Pre-Requisite:** Audio is supported by DUT and RTP/RTSP/TCP media streaming is implemented by DUT.

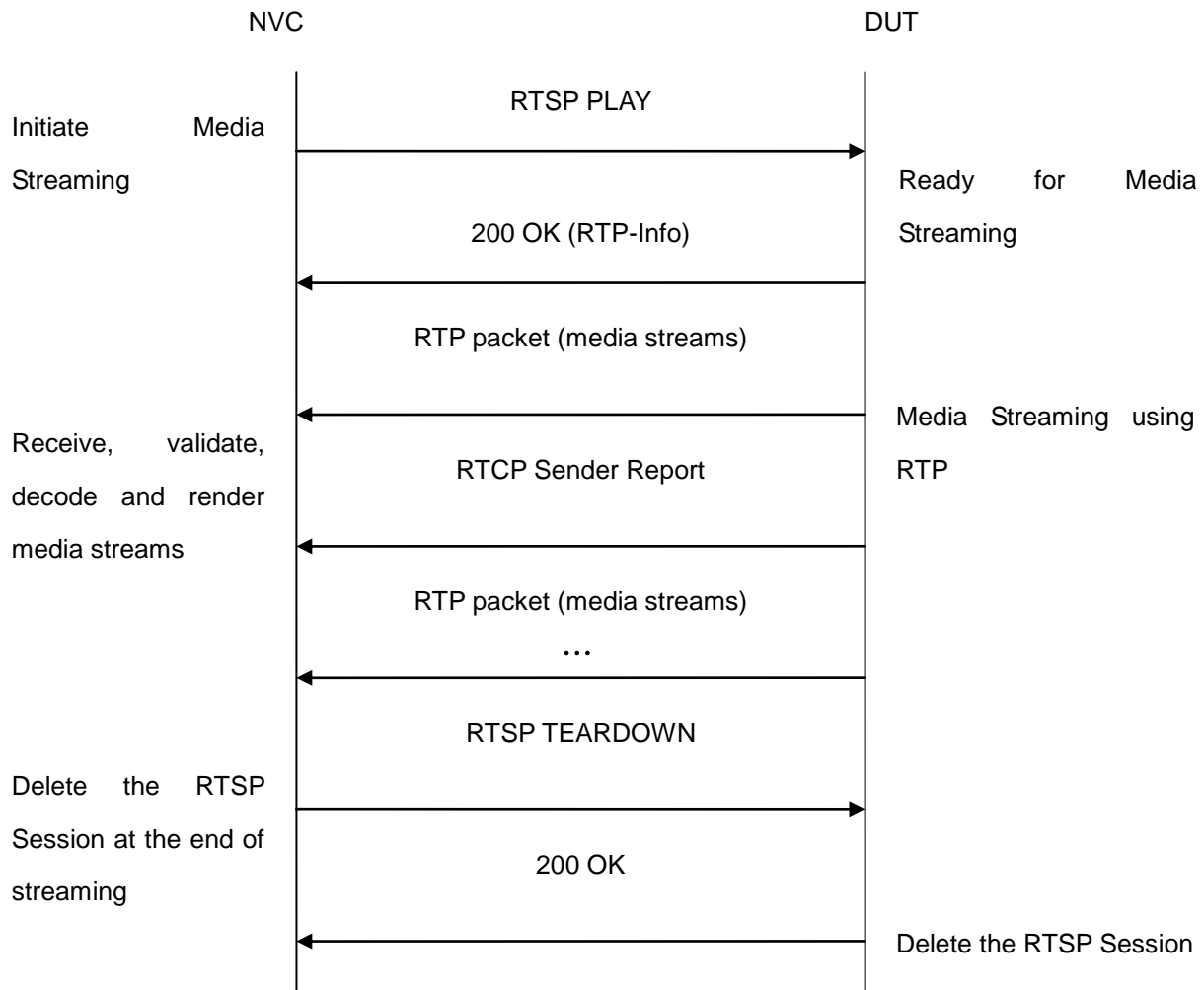
A media profile with JPEG video encoder configuration and G.711 audio encoder configuration

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and G.711 audio encoding support by following the procedure mentioned in Annex A.6.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "G711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.

8. NVC invokes GetStreamUriRequest message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.
9. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the RTSP media stream URI provided by the DUT.
11. NVC invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. NVC invokes RTSP SETUP request with transport parameter as 'RTP/TCP' along with 'interleaved' parameter for both audio and video streams separately.
14. DUT sends 200 OK message and the media stream information.
15. NVC invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
18. DUT validates the received RTP and RTCP packets, decodes and renders them.
19. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
20. DUT sends 200 OK Response and terminates the RTSP Session.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

#### **5.3.4 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/ UDP)**

**Test Label:** Real Time Viewing DUT JPEG/G.726 Audio&Video streaming using RTP-Unicast/UDP transport.

**Test Case ID:** RTSS-3-1-4

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (G.726)

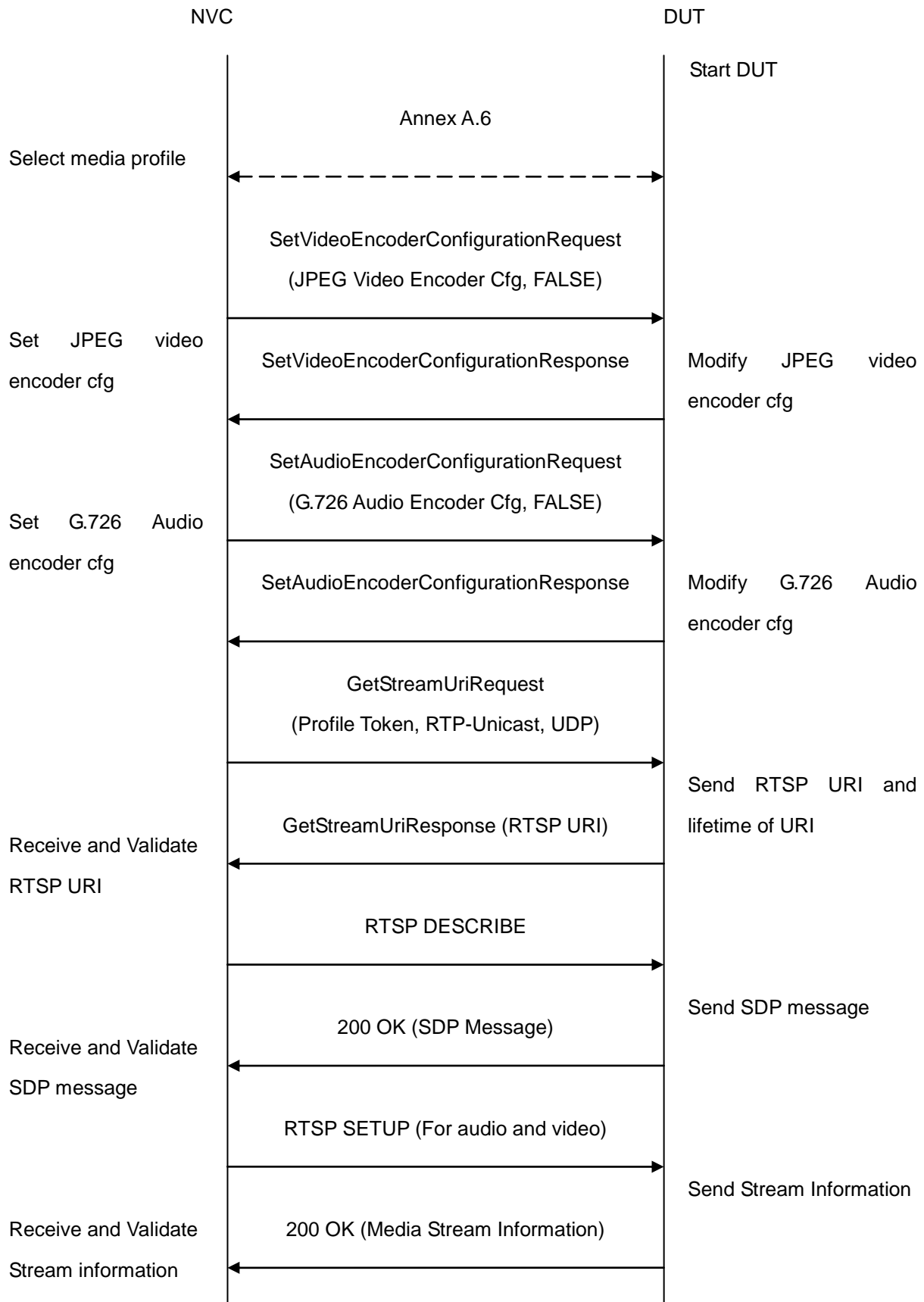
**Test Purpose:** To verify JPEG/G.726 Audio&Video streaming based on RTP/UDP Unicast Transport.

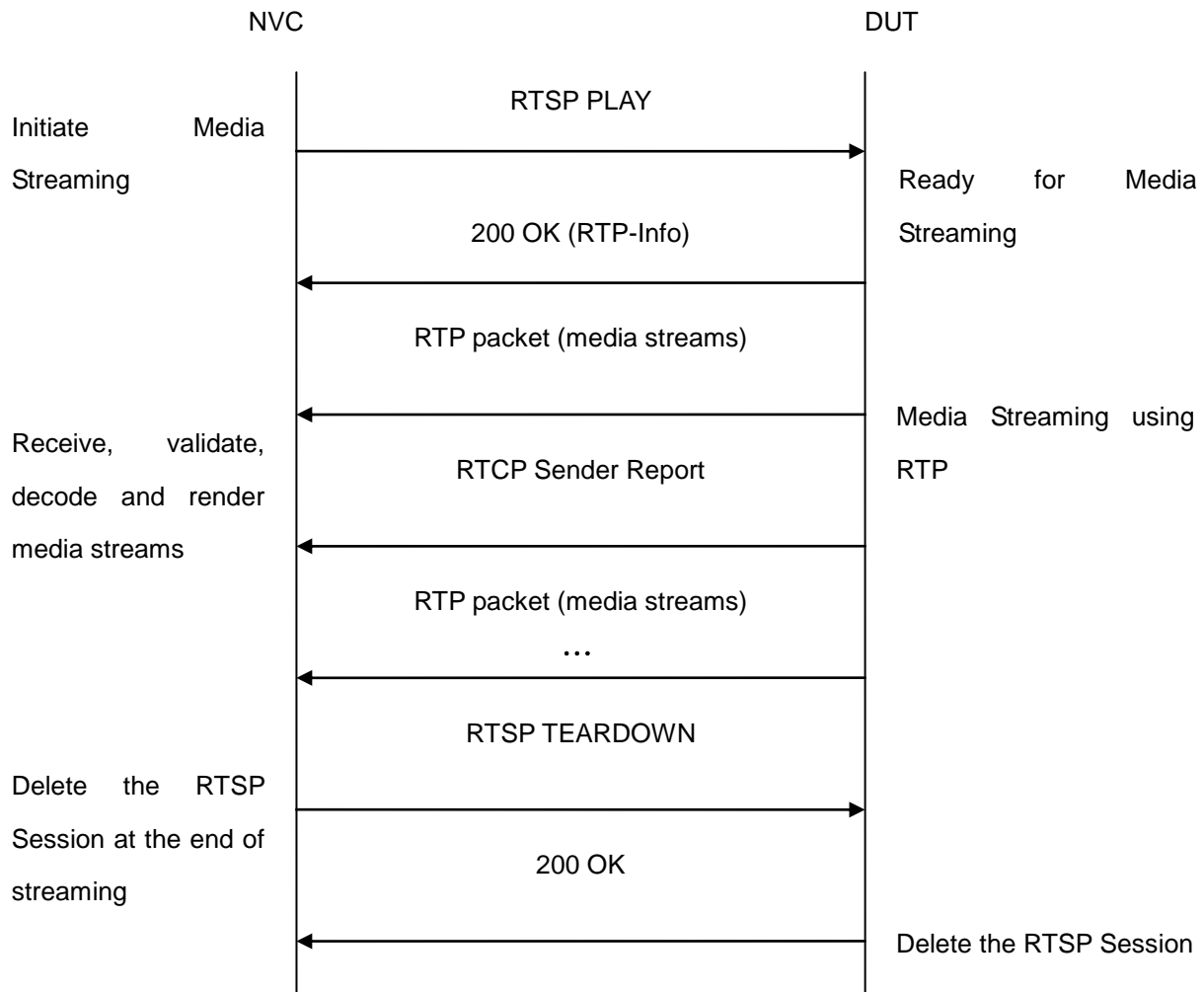
**Pre-Requisite:** Audio is supported by DUT and G.726 is implemented by DUT.

A media profile with JPEG video encoder configuration and G.726 audio encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and G.726 audio encoding support by following the procedure mentioned in Annex A.6.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "G726", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.

8. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the RTSP media stream URI provided by the DUT.
11. NVC invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. NVC invokes RTSP SETUP request with transport parameter as **RTP/UDP** for both audio and video streams separately.
14. DUT sends 200 OK message and the media stream information.
15. NVC invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT sends Audio/Video RTP media stream to NVC over UDP.
18. DUT sends Audio/Video RTCP sender report to NVC.
19. DUT validates the received RTP and RTCP packets, decodes and renders them.
20. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
21. DUT sends 200 OK Response and terminates the RTSP Session.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition

### **5.3.5 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP)**

**Test Label:** Real Time Viewing DUT JPEG/G.726 Audio&Video streaming using HTTP transport.

**Test Case ID:** RTSS-3-1-5

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (G.726)

**Test Purpose:** To verify JPEG/G.726 Audio&Video streaming based on HTTP Transport.

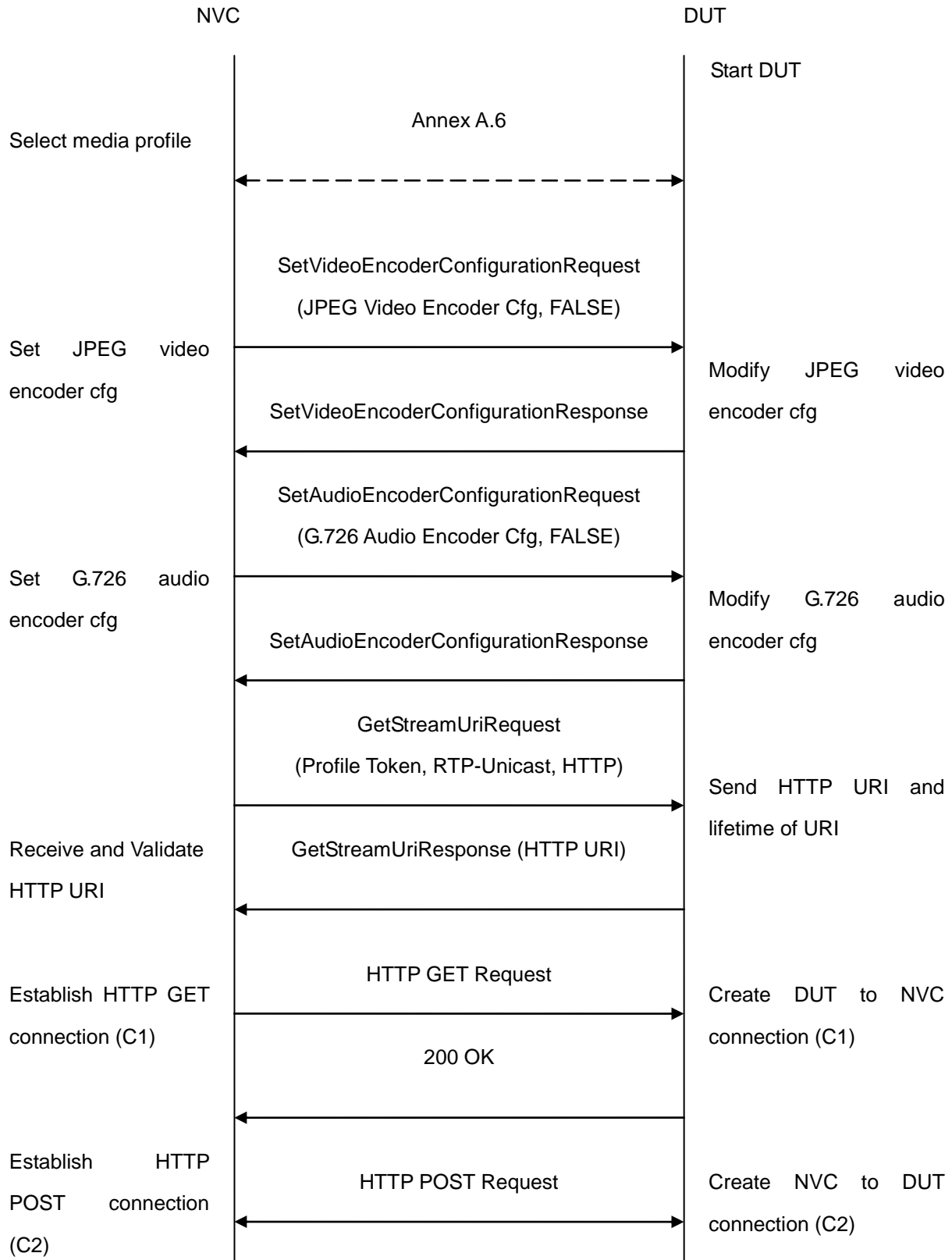
**Pre-Requisite:** Audio is supported by DUT and G.726 is implemented by DUT.

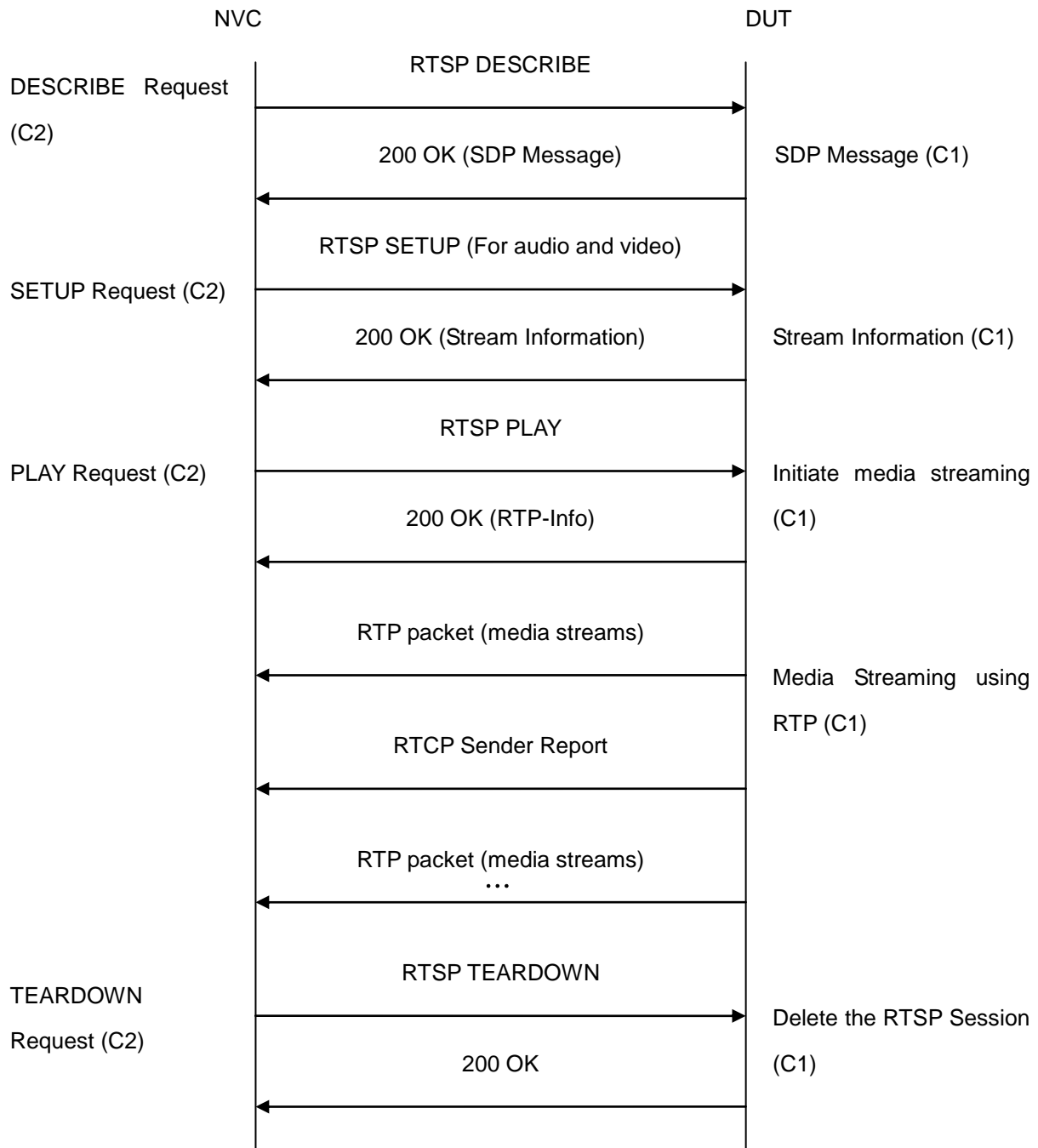
A media profile with JPEG video encoder configuration and G.726 audio encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**






**Test Procedure:**

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and G.726 audio encoding support by following the procedure mentioned in Annex A.6.

4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "G726", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
8. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, HTTP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends **HTTP URI** and parameters defining the lifetime of the URI like **ValidUntilConnect, ValidUntilReboot and Timeout** in the GetStreamUriResponse message.
10. NVC verifies the HTTP media stream URI provided by the DUT.
11. NVC invokes HTTP GET Request on DUT and establishes DUT to NVC connection for RTP data transfer.
12. NVC invokes HTTP POST Request and establishes NVC to DUT connection for RTSP control requests.
13. NVC invokes RTSP DESCRIBE request on HTTP POST connection.
14. DUT sends 200 OK message and SDP information on HTTP GET connection.
15. NVC invokes RTSP SETUP requests on HTTP POST connection with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter for both audio and video streams separately.
16. DUT sends 200 OK message and the media stream information on HTTP GET connection.
17. NVC invokes RTSP PLAY request on HTTP POST connection.
18. DUT sends 200 OK message and starts media streaming on HTTP GET connection.
19. DUT transfers Audio/Video RTP media stream to NVC on HTTP GET connection.
20. DUT sends Audio/Video RTCP sender report to NVC on HTTP GET connection.
21. DUT validates the received RTP and RTCP packets, decodes and renders them.
22. NVC invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.
23. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

#### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.3.6 MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP)

**Test Label:** Real Time Viewing DUT JPEG/G.726 Audio&Video streaming using RTP/RTSP/TCP transport.

**Test Case ID:** RTSS-3-1-6

**ONVIF Core Specification Coverage:** RTP/RTSP/TCP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (G.726 & RTP/RTSP/TCP)

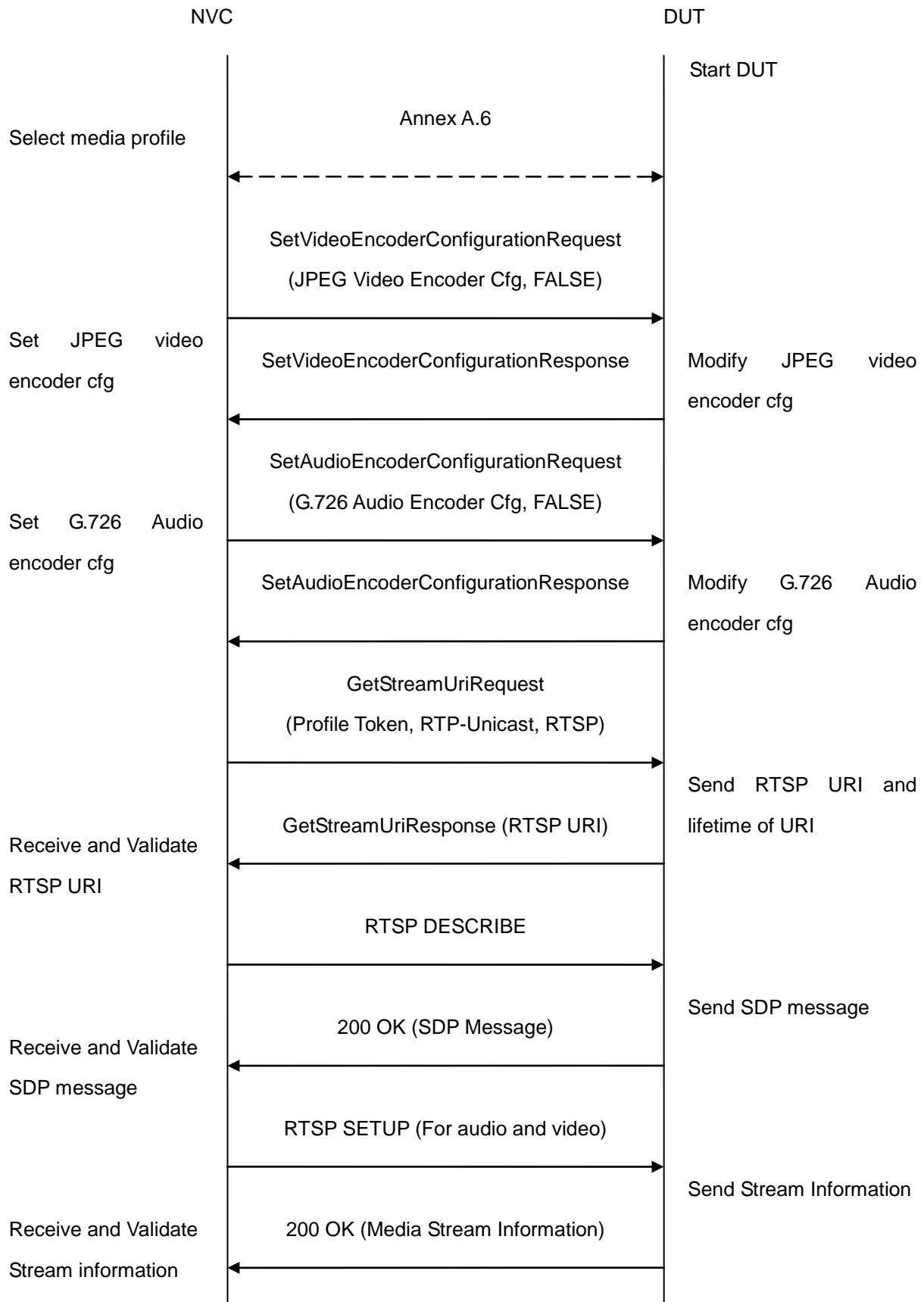
**Test Purpose:** To verify JPEG/G.726 Audio&Video streaming based on RTP/RTSP/TCP using RTSP tunnel.

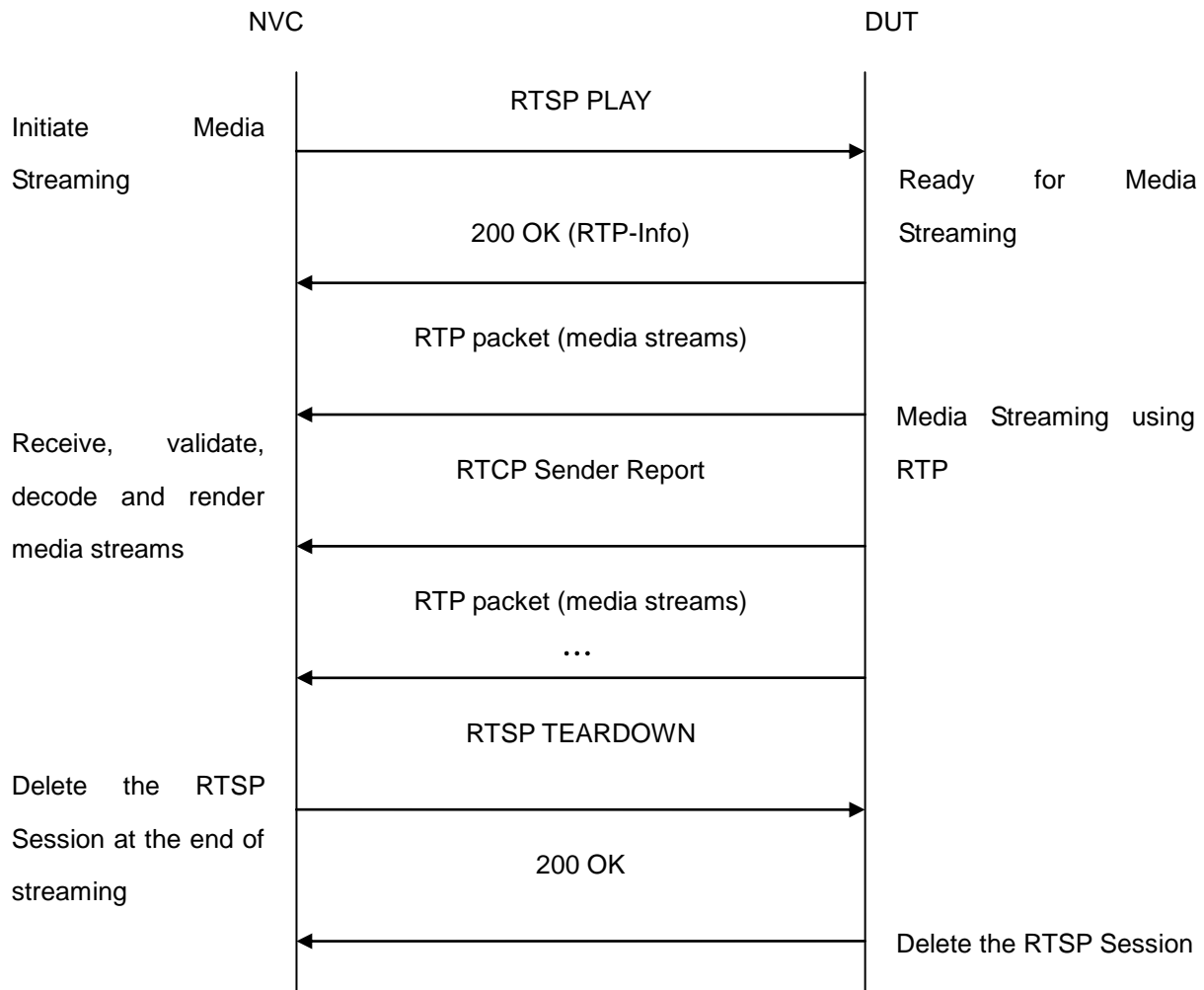
**Pre-Requisite:** Audio is supported by DUT, G.726 and RTP/RTSP/TCP media streaming is implemented by DUT.

A media profile with JPEG video encoder configuration and G.726 audio encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and G.726 audio encoding support by following the procedure mentioned in Annex A.6.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "G726", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.

8. NVC invokes GetStreamUri request (**Profile Token, RTP-Unicast, RTSP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the RTSP media stream URI provided by the DUT.
11. NVC invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. NVC invokes RTSP SETUP request with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter for both audio and video streams separately.
14. DUT sends 200 OK message and the media stream information.
15. NVC invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
18. DUT validates the received RTP and RTCP packets, decodes and renders them.
19. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
20. DUT sends 200 OK Response and terminates the RTSP Session.

**Test Result:****PASS –**

DUT passes all assertions.

**FAIL –**

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

### 5.3.7 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/ UDP)

**Test Label:** Real Time Viewing DUT JPEG/AAC Audio&Video streaming using RTP-Unicast/UDP transport.

**Test Case ID:** RTSS-3-1-7

**ONVIF Core Specification Coverage:** RTP data transfer via UDP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (AAC)

**Test Purpose:** To verify JPEG/AAC Audio&Video streaming based on RTP/UDP Unicast Transport.

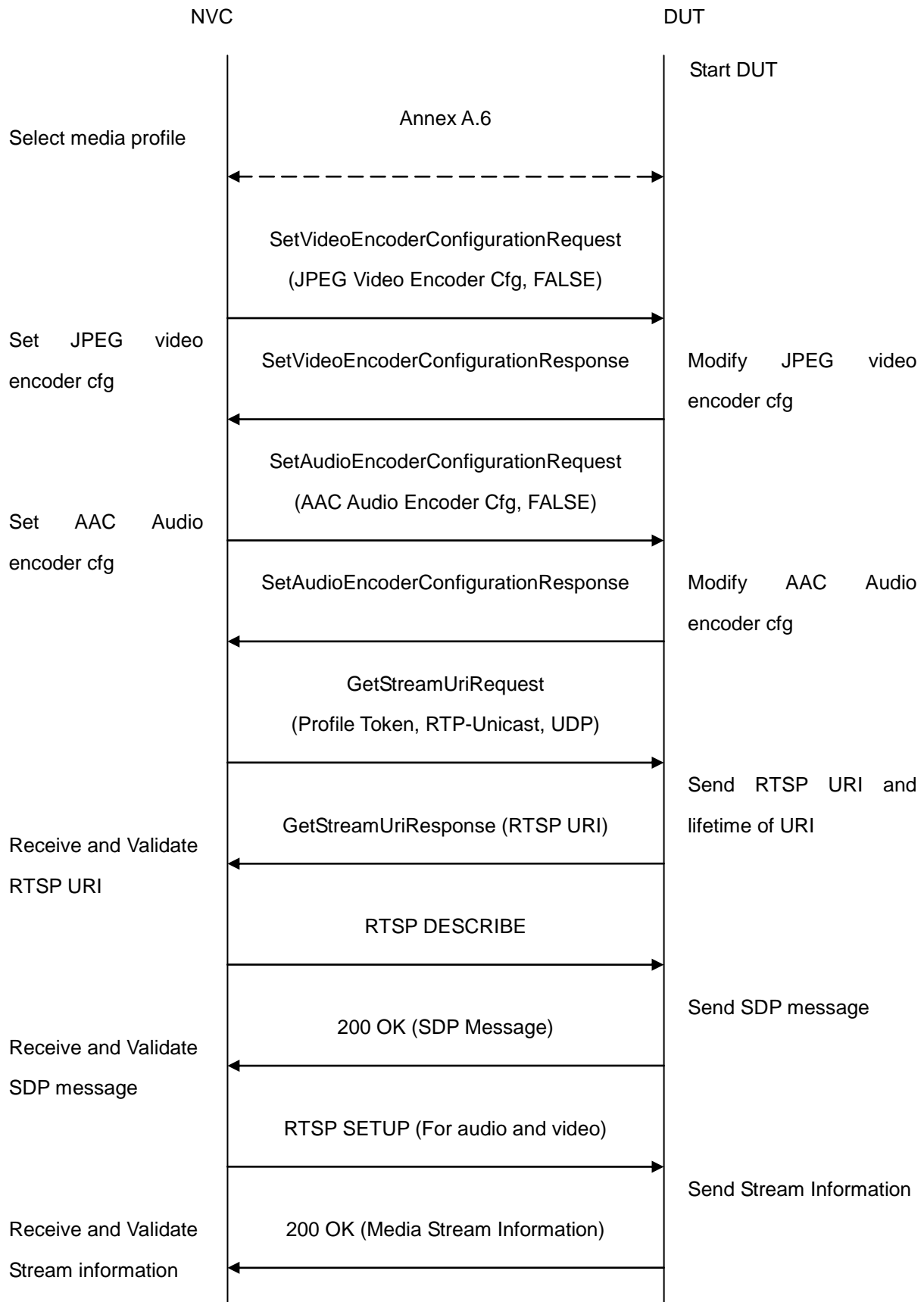
**Pre-Requisite:** Audio is supported by DUT and AAC is implemented by DUT.

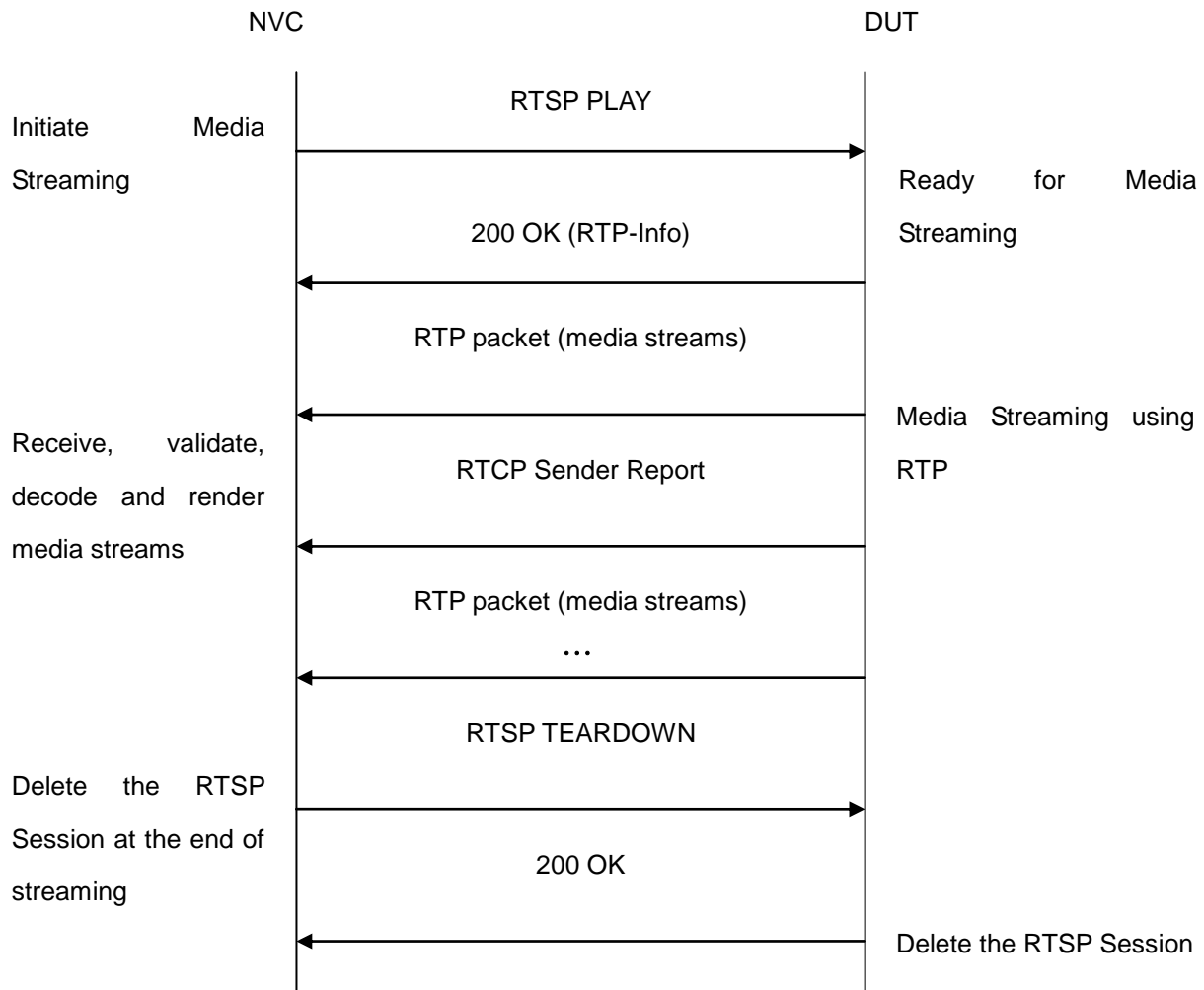
A media profile with JPEG video encoder configuration and AAC audio encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and AAC audio encoding support by following the procedure mentioned in Annex A.6.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "AAC", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.

8. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the RTSP media stream URI provided by the DUT.
11. NVC invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. NVC invokes RTSP SETUP requests for audio and video streams separately with transport parameter as **RTP/UDP** for both audio and video streams separately.
14. DUT sends 200 OK message and the media stream information.
15. NVC invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT sends Audio/Video RTP media stream to NVC over UDP.
18. DUT sends Audio/Video RTCP sender report to NVC.
19. DUT validates the received RTP and RTCP packets, decodes and renders them.
20. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
21. DUT sends 200 OK Response and terminates the RTSP Session.

#### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.3.8 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP)

**Test Label:** Real Time Viewing DUT JPEG/AAC Audio&Video streaming using HTTP transport.

**Test Case ID:** RTSS-3-1-8

**ONVIF Core Specification Coverage:** RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (AAC)

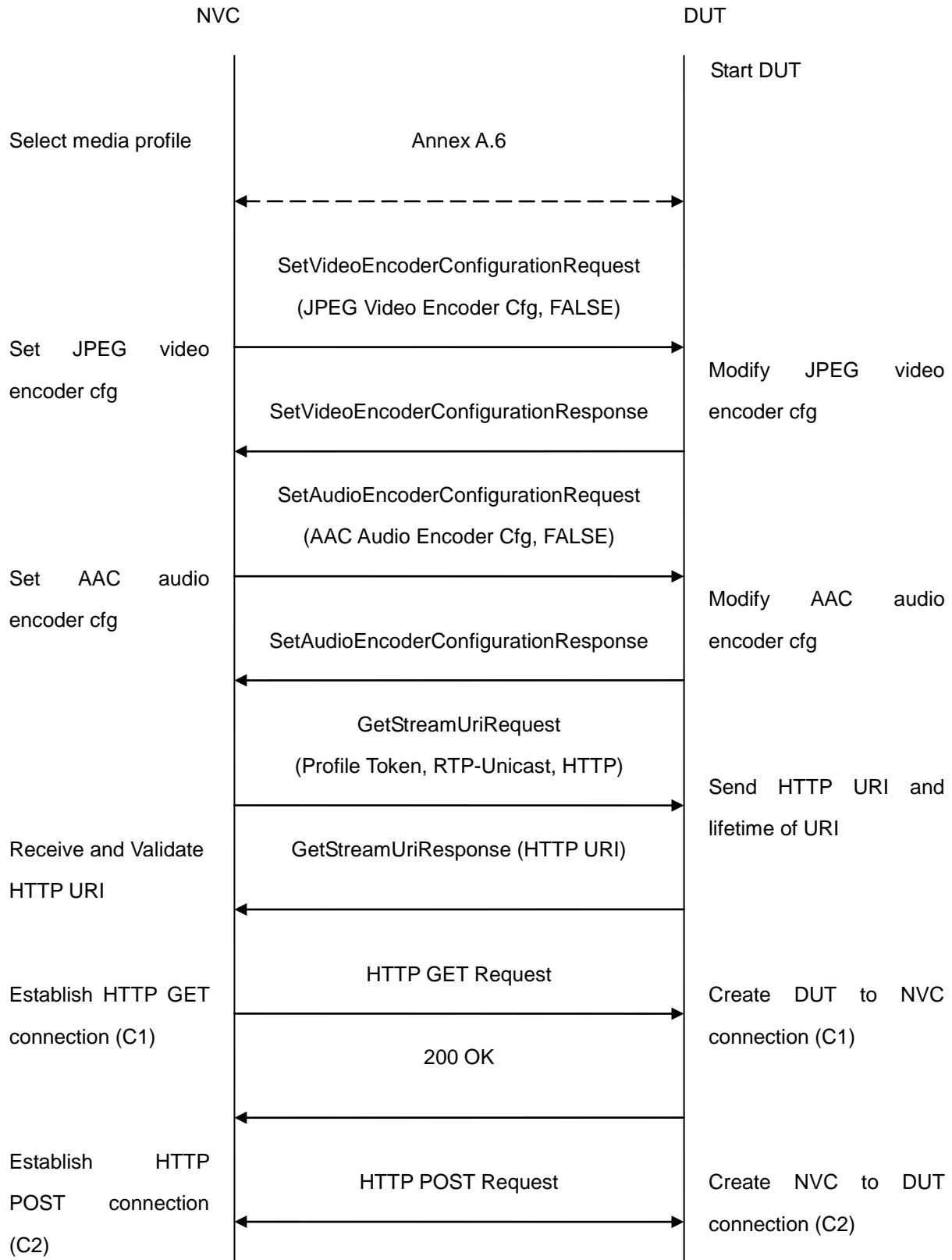
**Test Purpose:** To verify JPEG/AAC Audio&Video streaming based on HTTP Transport.

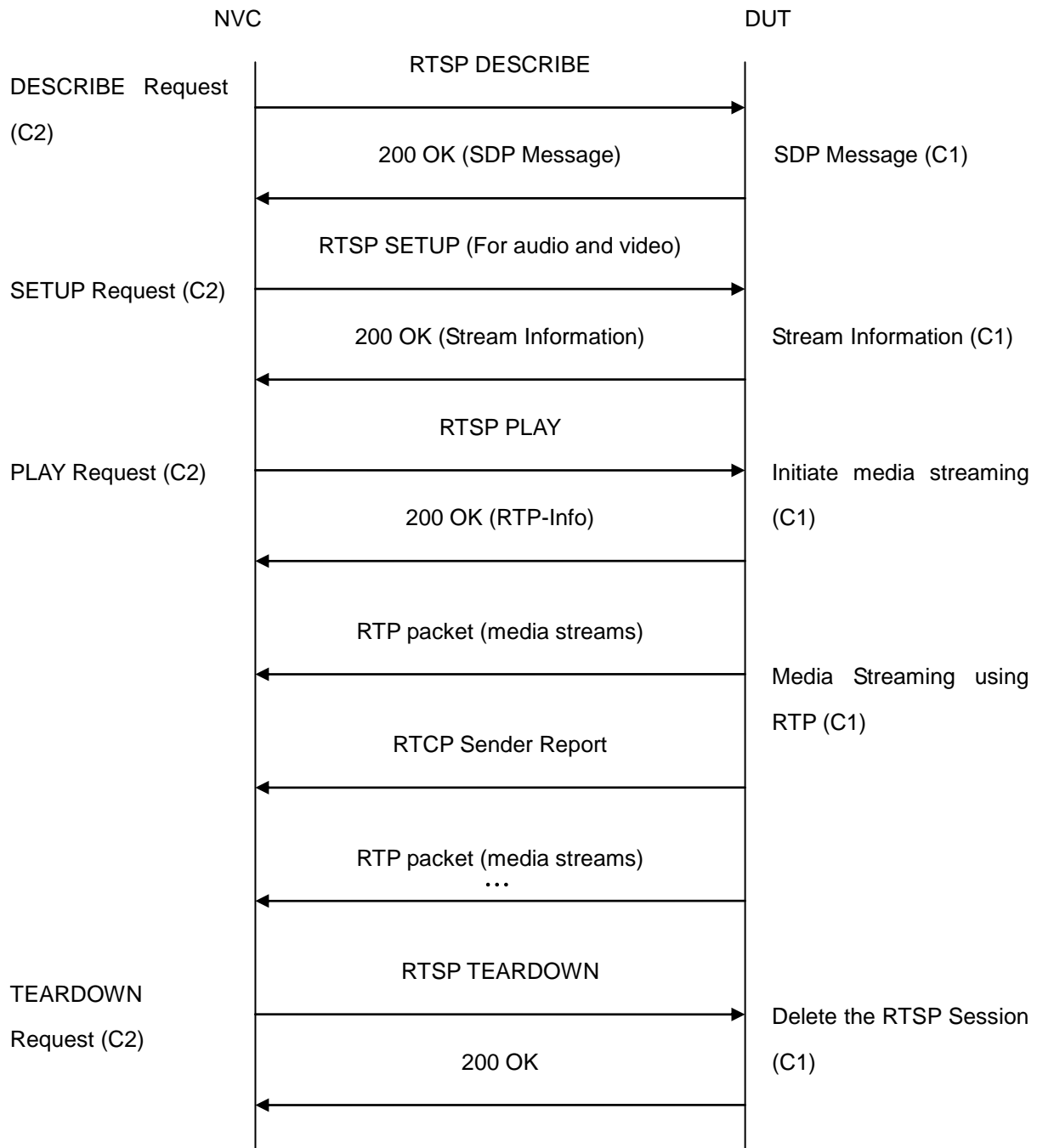
**Pre-Requisite:** Audio is supported by DUT and AAC is implemented by DUT.

A media profile with JPEG video encoder configuration and AAC audio encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**





#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and AAC audio encoding support by following the procedure mentioned in Annex A.6.

4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "AAC", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
8. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, HTTP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the HTTP media stream URI provided by the DUT.
11. NVC invokes HTTP GET Request on DUT and establishes DUT to NVC connection for RTP data transfer.
12. NVC invokes HTTP POST Request and establishes NVC to DUT connection for RTSP control requests.
13. NVC invokes RTSP DESCRIBE request on HTTP POST connection.
14. DUT sends 200 OK message and SDP information on HTTP GET connection.
15. NVC invokes RTSP SETUP request on HTTP POST connection with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter for both audio and video streams separately.
16. DUT sends 200 OK message and the media stream information on HTTP GET connection.
17. NVC invokes RTSP PLAY request on HTTP POST connection.
18. DUT sends 200 OK message and starts media streaming on HTTP GET connection.
19. DUT transfers Audio/Video RTP media stream to NVC on HTTP GET connection.
20. DUT sends Audio/Video RTCP sender report to NVC on HTTP GET connection.
21. DUT validates the received RTP and RTCP packets, decodes and renders them.
22. NVC invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.
23. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

#### Test Result:

##### PASS –

DUT passes all assertions.

##### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send valid RTP header in one or more media streams.

DUT did not send RTCP sender report correctly.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

See Annex A.2 for Invalid RTP header definition.

### 5.3.9 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP)

**Test Label:** Real Time Viewing DUT JPEG/AAC Audio&Video streaming using RTP/RTSP/TCP transport.

**Test Case ID:** RTSS-3-1-9

**ONVIF Core Specification Coverage:** RTP/RTSP/TCP, RTP, RTCP, Stream control, RTSP.

**Command Under Test:** None

**WSDL Reference:** None

**Requirement Level:** MUST IF SUPPORTED (Audio) & IMPLEMENTED (AAC & RTP/RTSP/TCP)

**Test Purpose:** To verify JPEG/AAC Audio&Video streaming based on RTP/RTSP/TCP using RTSP tunnel.

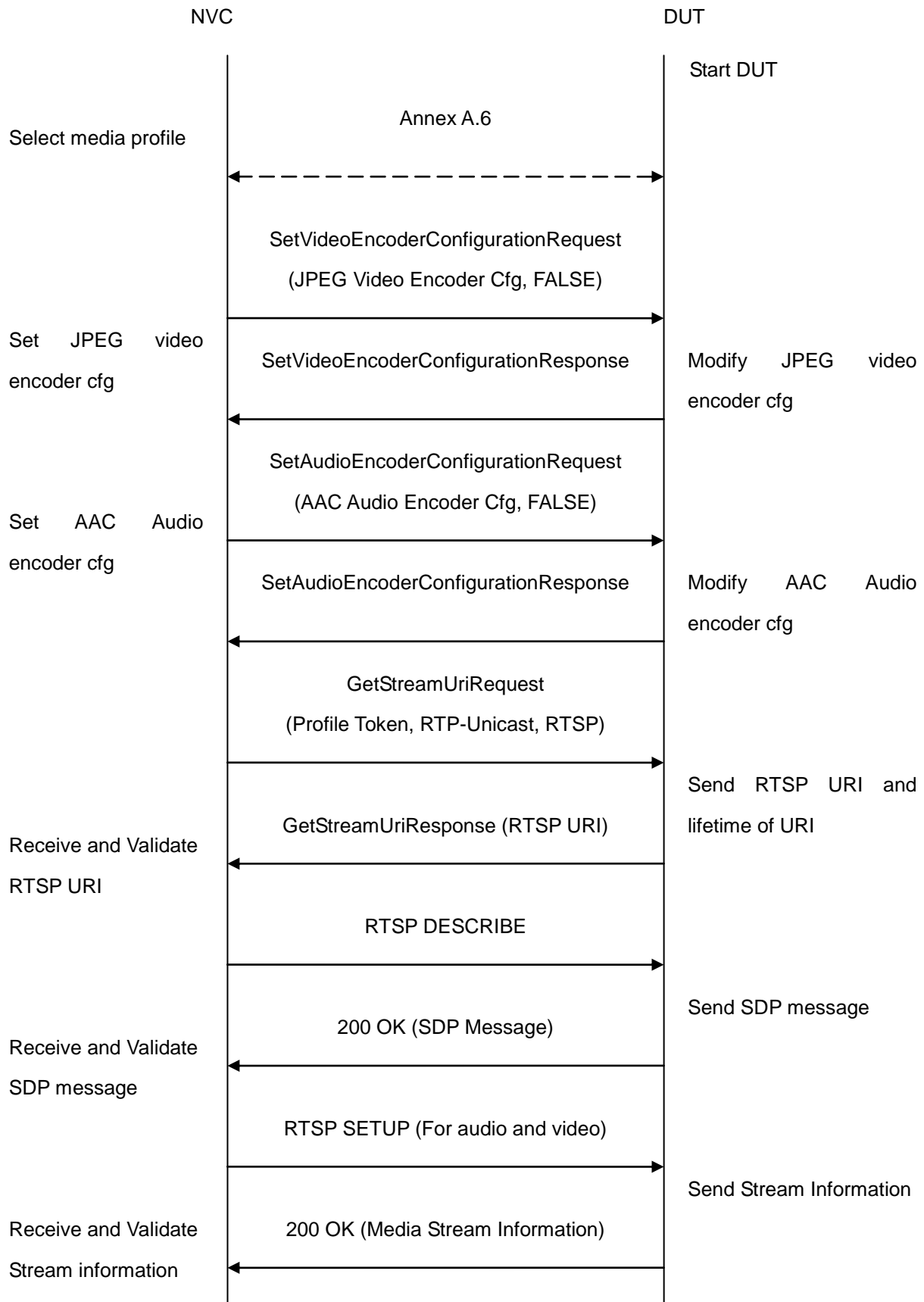
**Pre-Requisite:** Audio is supported by DUT, AAC and RTP/RTSP/TCP media streaming is implemented by DUT.

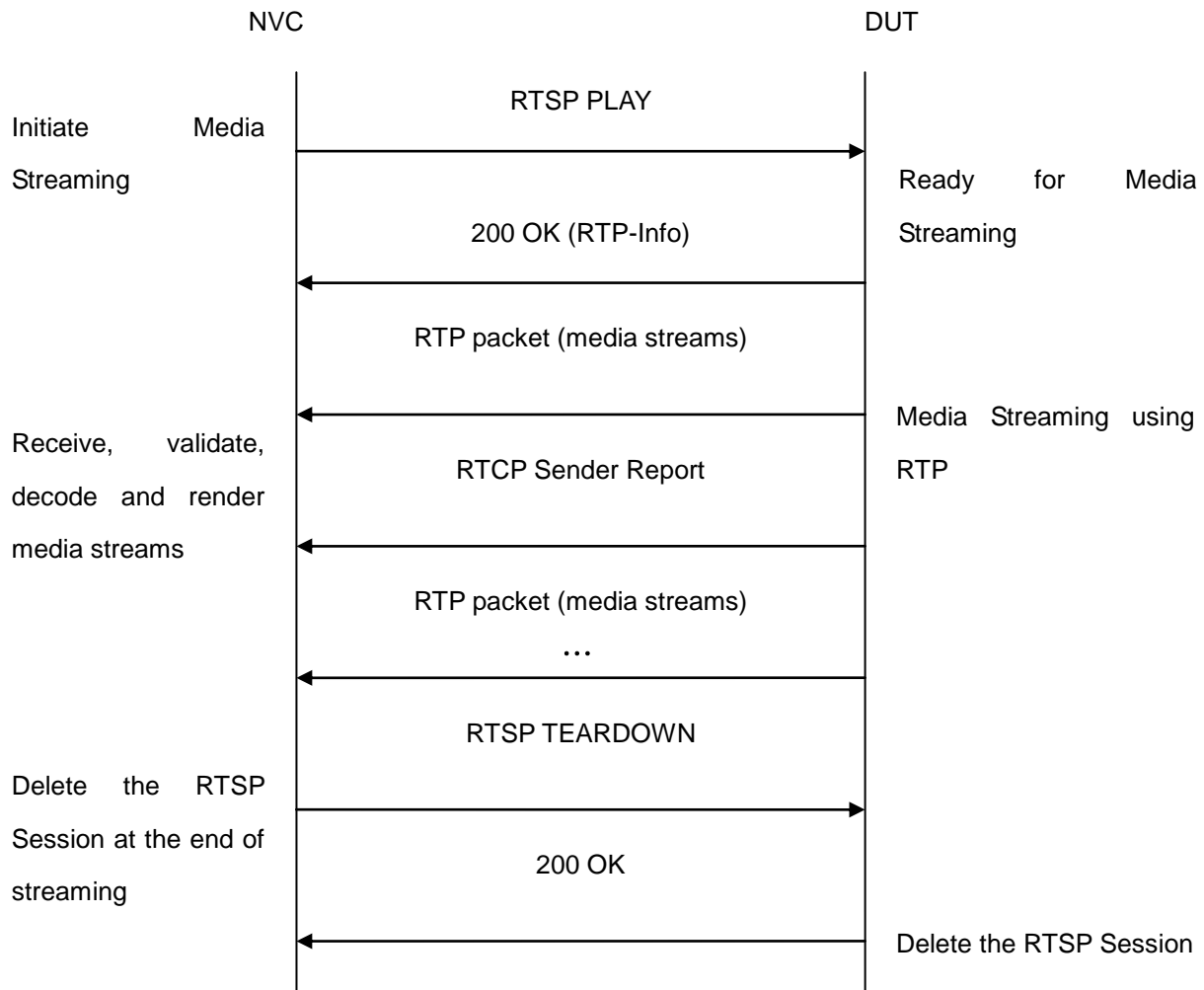
A media profile with JPEG video encoder configuration and AAC audio encoder configuration.

**Test Configuration:** NVC and DUT

**Test Sequence:**







#### Test Procedure:

1. Start an NVC.
2. Start an DUT.
3. NVC selects a media profile with both JPEG video encoding support and AAC audio encoding support by following the procedure mentioned in Annex A.6.
4. NVC invokes SetVideoEncoderConfigurationRequest (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetVideoEncoderConfigurationOptions response in A.6.
5. DUT modifies video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
6. NVC invokes SetAudioEncoderConfigurationRequest (**Encoding = "AAC", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false**). These values will be taken from the GetAudioEncoderConfigurationOptions response in A.6.
7. DUT modifies audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.

8. NVC invokes GetStreamUriRequest message (**Profile Token, RTP-Unicast, RTSP transport**) to retrieve media stream URI for the selected media profile.
9. DUT sends RTSP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetStreamUriResponse message.
10. NVC verifies the RTSP media stream URI provided by the DUT.
11. NVC invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. NVC invokes RTSP SETUP request with transport parameter as '**RTP/TCP**' along with '**interleaved**' parameter for both audio and video streams separately.
14. DUT sends 200 OK message and the media stream information.
15. NVC invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
18. DUT validates the received RTP and RTCP packets, decodes and renders them.
19. NVC invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
20. DUT sends 200 OK Response and terminates the RTSP Session.

#### Test Result:

#### PASS –

DUT passes all assertions.

#### FAIL –

DUT did not have valid media profile which has both audio and video encoder configurations.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetStreamUriResponse message.

DUT did not send one or more mandatory parameters in the GetStreamUriResponse message (mandatory parameters – RTSP URI, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.

DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.

RTSP Session is terminated by DUT during media streaming.

**Note:** See Annex A.3 for correct syntax for the StreamSetup element in GetStreamUri requests.

## 5.4 Notification Streaming Interface

### 5.4.1 NOTIFICATION STREAMING

**Test Label:** event handling Notification Streaming

**Test Case ID:** RTSS-4-1-1

<b>ONVIF</b>	<b>Core</b>	<b>Specification</b>	<b>Coverage:</b>	CreateProfile,
GetVideoSourceConfigurations, GetMetadataConfigurations,			AddVideoSourceConfiguration,	
AddMetadataConfiguration, SetMetadataConfiguration,			GetStreamUri, SetSynchronizationPoint,	
DeleteProfile, Notification Streaming Interface				

**Command Under Test:** CreateProfile, GetVideoSourceConfigurations, GetMetadataConfigurations, AddVideoSourceConfiguration, AddMetadataConfiguration, SetMetadataConfiguration, GetStreamUri, SetSynchronizationPoint, DeleteProfile

**WSDL Reference:** media.wsdl

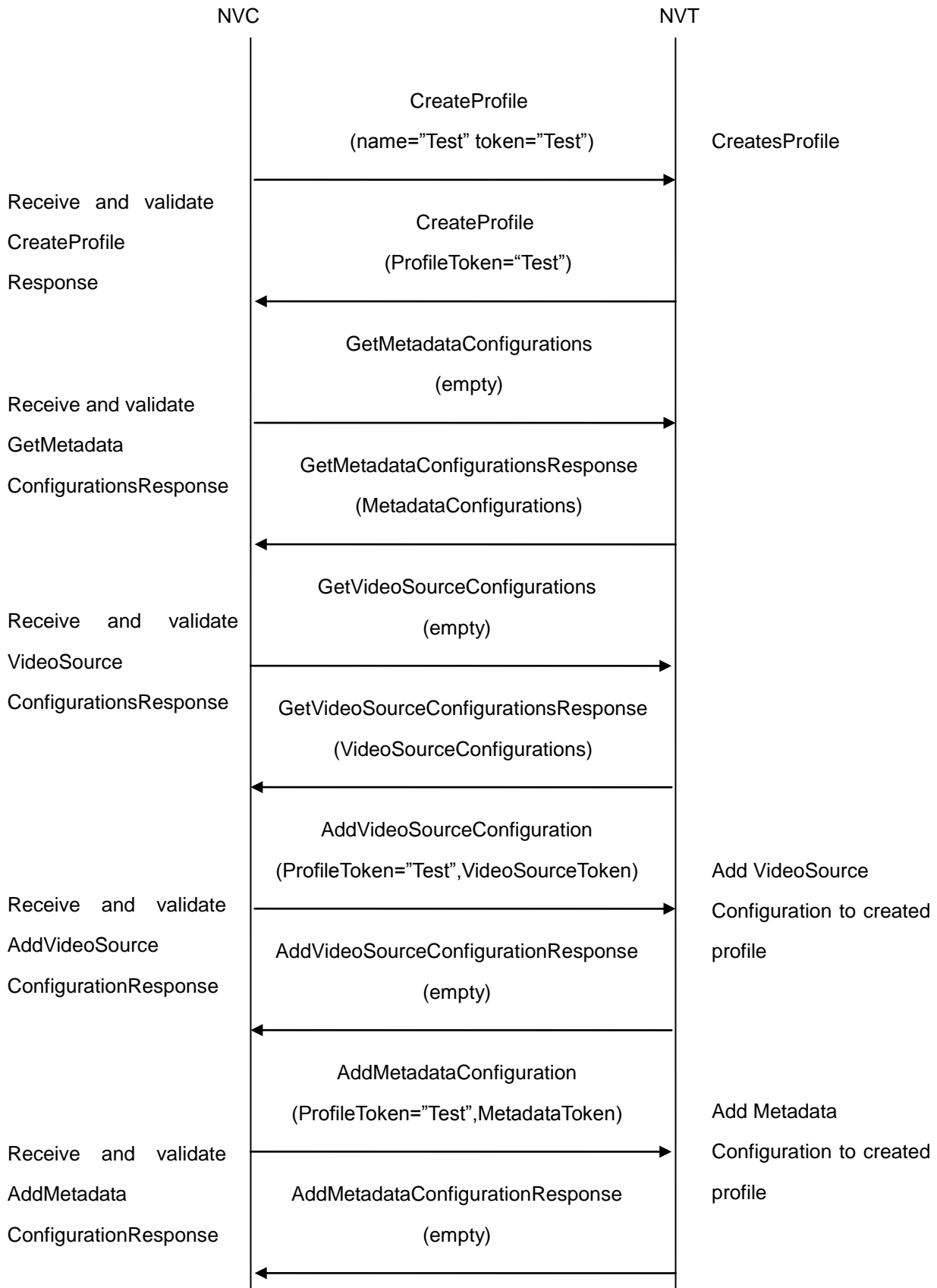
**Requirement Level:** MUST

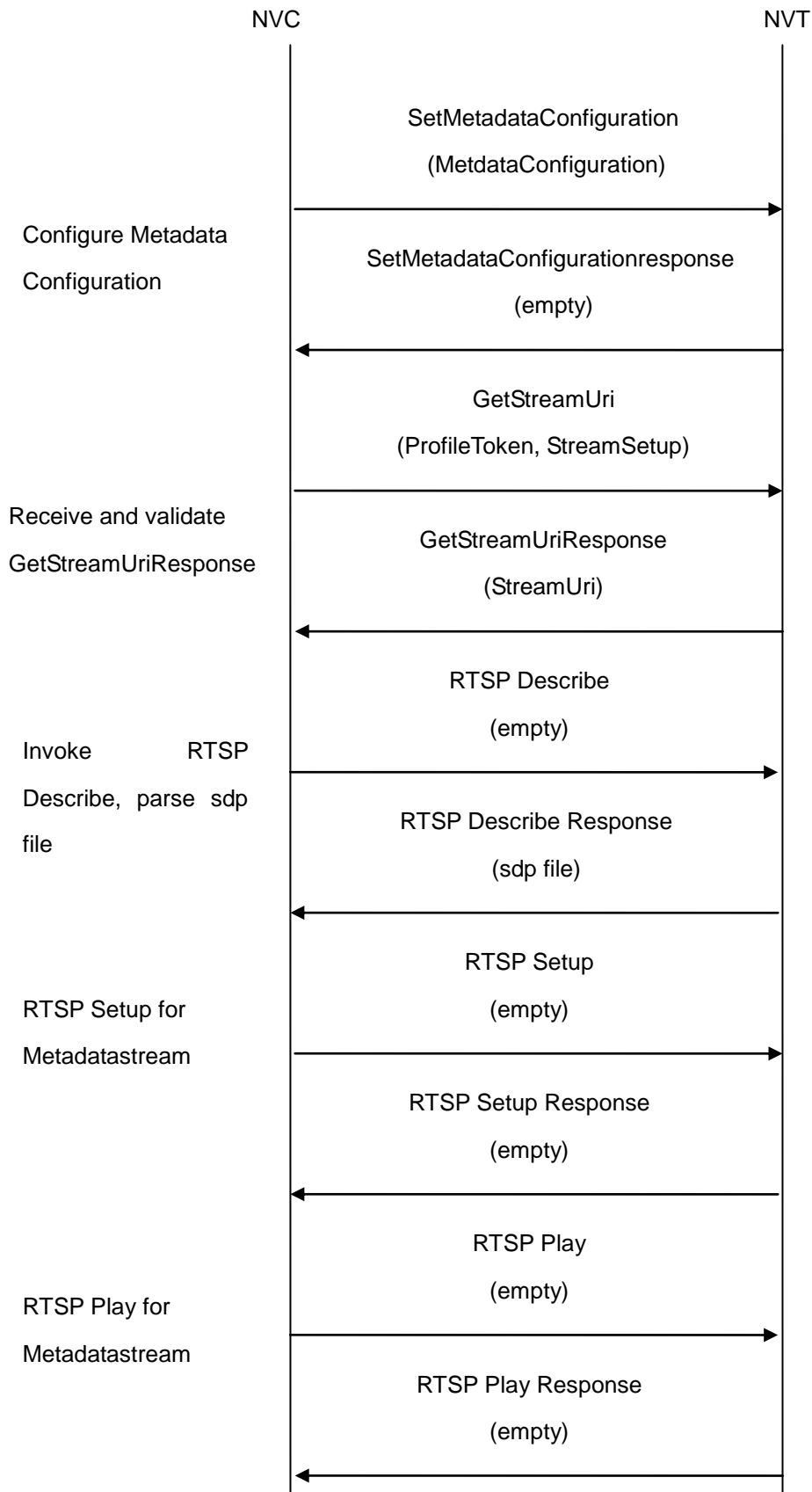
**Test Purpose:** To verify Notification Streaming

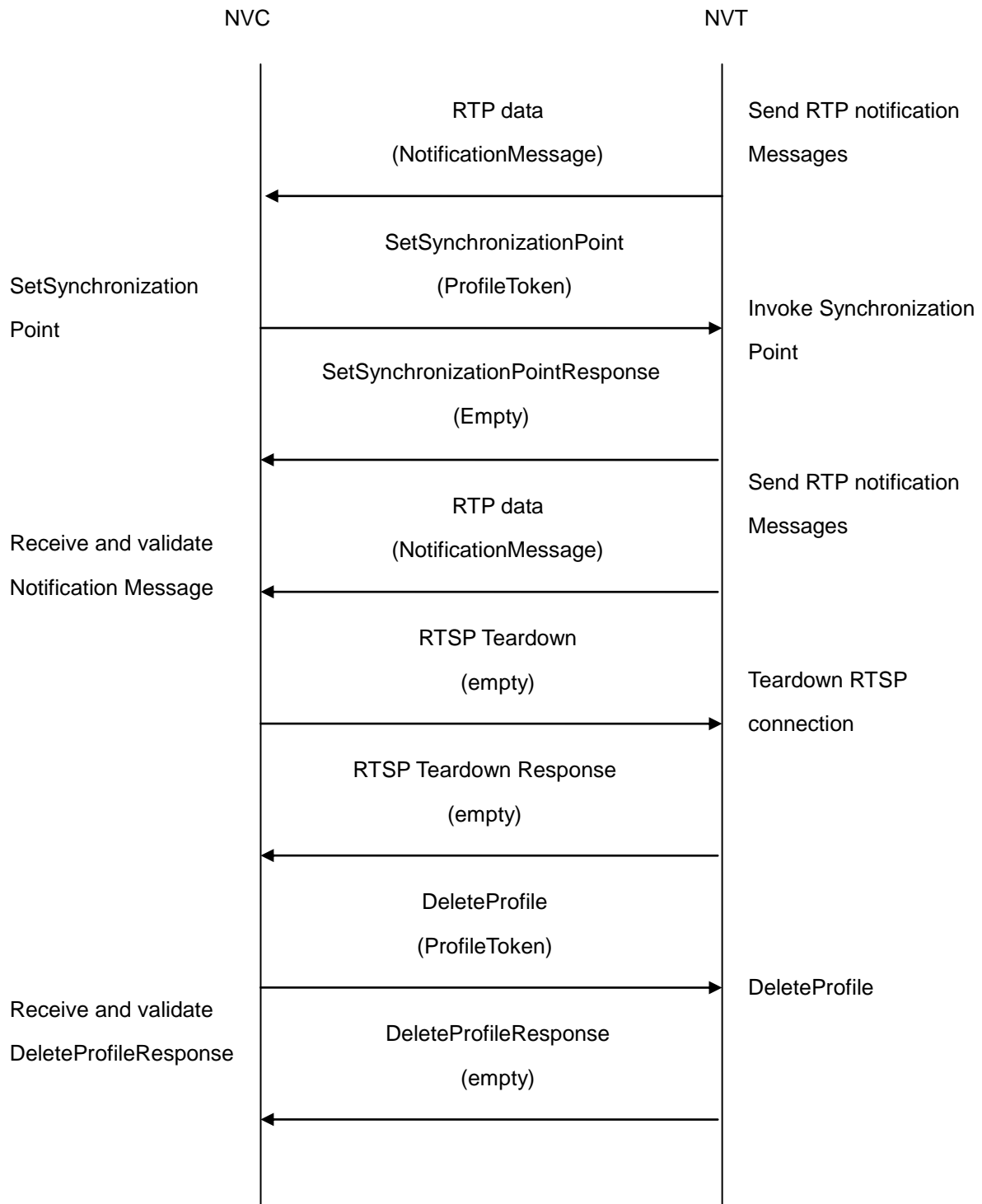
**Pre-Requisite:** The device needs to provide at least one topic representing a certain property. If the device does not support a property event the vendor MUST make sure that another event is sent during testing.

**Test Configuration:** NVC and DUT

**Test Sequence:**







**Test Procedure:**

1. Start an NVC.
2. Start an DUT.

3. NVC invokes CreateProfile (name="Test", ProfileToken="Test") to create a new empty profile that is used for this test scenario.
4. Verify that the DUT sends a valid CreateProfileResponse
5. NVC will invoke GetMetadataConfigurations to retrieve all existing MetadataConfigurations of the device
6. Verify that the DUT sends a valid GetMetadataConfigurationResponse (that contains at least one MetadataConfiguration).
7. NVC will invoke GetVideoSourceConfigurations
8. Verify that the DUT sends a valid GetVideoSourceConfigurationResponse
9. NVC will select a VideoSourceConfiguration and add this configuration to the created profile
10. Verify that the DUT sends a valid AddVideoSourceConfigurationResponse
11. NVC will select a MetadataConfiguration and add this configuration to the created profile
12. Verify that the DUT sends a valid AddMetadataConfigurationResponse
13. NVC will invoke SetMetadataConfiguration(<Analytics>false</Analytics>,<Events/>) to configure the Metadatastream; The NVC is interested in receiving all events, therefore no Filter is applied. For details on the usage of the MetadataConfiguration elements, see Annex A.7.
14. Verify that the DUT sends a valid SetMetadataConfigurationResponse
15. NVC will invoke GetStreamUri (ProfileToken, RTP-Unicast)
16. Verify that the DUT sends a GetStreamUriResponse including a valid StreamUri
17. NVC will invoke RTSP Describe to retrieve the sdp file
18. Verify that DUT sends a 200 OK Response
19. Validate sdp file (sdp file contains only one media section; rtpmap=vnd.onvif.metadata)
20. NVC will invoke RTSP Setup for the Metadatastream
21. Verify that the DUT send a 200 OK Response
22. NVC will invoke RTSP Play
23. Verify that the DUT sends a 200 OK Response
24. Receive and validate RTP Notification messages
25. NVC will invoke the SetSynchronizationPoint command to trigger events; if the device does not support property events the vendor MUST trigger the events manually.
26. Validate that DUT sends a valid SetSynchronizationPointResponse
27. Verify that at least one RTP Notification is sent.
28. Receive and validate RTP Notification messages and check that the PropertyOperation is "Initialized" or "Changed" if it is an Property event
29. NVC will invoke RTSP Teardown to terminate the RTSP session



30. Verify that DUT sends a 200 OK Response

31. NVC will invoke DeleteProfile

32. Verify that DUT sends a DeleteProfileResponse

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send a CreateProfileResponse

The DUT did not send a valid CreateProfileResponse

The DUT did not send valid GetMetadataConfigurationsResponse; At least one MetadataConfiguration MUST be present

The DUT did not send a valid GetVideoSourceConfigurationsResponse

The DUT did not send a valid AddVideoSourceConfigurationResponse

The DUT did not send a valid AddMetadataConfigurationResponse

The DUT did not send a valid SetMetadataConfigurationResponse

The DUT did not send a GetStreamUriResponse including a valid StreamUri

The DUT did not send a 200 OK RTSP DESCRIBE Response

The DUT did not send a valid sdp file

The DUT did not send a 200 OK RTSP SETUP Response

The DUT did not send a 200 OK RTSP PLAY Response

The DUT did not send RTP data

The DUT did not send a SetSynchronizationPointResponse

The DUT did not send at least one event

The DUT did not send RTP data with PropertyOperation="Initialized" or "Changed" if it is an property event

The DUT did not send a 200 OK RTSP TEARDOWN Response

The DUT did not send a DeleteProfileResponse

## 6 Annex

This section describes the meaning of the following definitions. These definitions are used in the test case description.

### A.1 Invalid Media Profile

Media profile token is a string of max length value of 64.

Invalid Media Profile:

A string which is not formed according to the rules of RFC 952.

A string which exceeds max length value of 64.

### A.2 Invalid RTP Header

A RTP header which is not formed according to the header field format defined in the RFC 3550 Section 5.1 is considered as invalid RTP header.

### A.3 StreamSetup syntax for GetStreamUri

The following media stream setups for GetStreamUri are covered in this Test Specification:

1. RTP unicast over UDP
2. RTP over RTSP over HTTP over TCP
3. RTP over RTSP over TCP

The correct syntax for the StreamSetup element for these media stream setups are as follows:

1. RTP unicast over UDP

```
<StreamSetup>
  <StreamType>RTP_unicast</StreamType>
  <Transport>
    <Protocol>UDP</Protocol>
  </Transport>
</StreamSetup>
```

2. RTP over RTSP over HTTP over TCP

```
<StreamSetup>
  <StreamType>RTP_unicast</StreamType>
  <Transport>
    <Protocol>HTTP</Protocol>
  </Transport>
</StreamSetup>
```

### 3. RTP over RTSP over TCP

```
<StreamSetup>  
  <StreamType>RTP_unicast</StreamType>  
  <Transport>  
    <Protocol>RTSP</Protocol>  
  </Transport>  
</StreamSetup>
```

#### A.4 I-frame insertion time interval

‘I-frame insertion time interval’ is the time interval between two consecutive I-frames sent by DUT.

NVC calculates this value by using ‘GovLength’ parameter in the Video encoder configuration. NVC has to configure ‘GovLength’ to larger value so that there will be sufficient time difference between two I-frames.

In case of SetSynchronizationPoint test cases in “Real Time Streaming” section, NVC follows the following procedure to verify that I-frame is inserted as a result of “SetSynchronizationPointRequest”.

NVC waits for an I-frame before invoking SetSynchronizationPointRequest.

After receiving I-frame, NVC starts a timer with time out period less than ‘I-frame insertion time interval’ and immediately invokes SetSynchronizationPointRequest.

NVC waits for the I-frame and verifies that it receives I-frame before the timeout period.

## A.5 Media Profile Configuration for Video Streaming

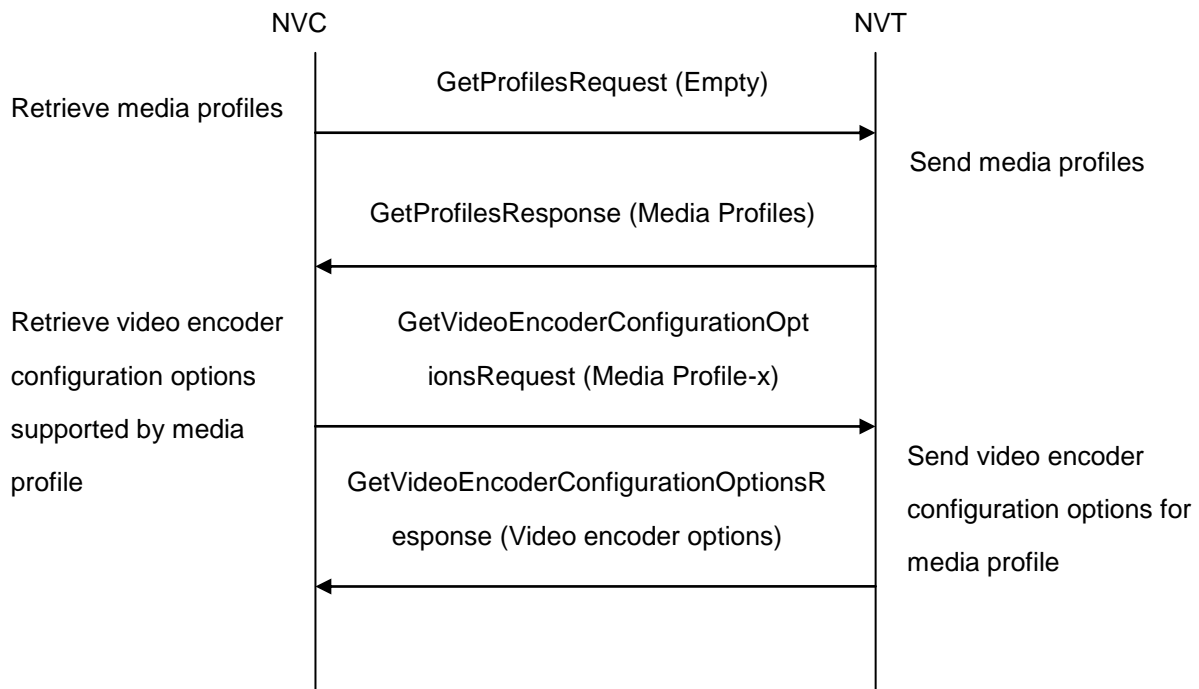
For the execution of real time streaming - video test cases, NVC has to select and configure the media profile based on the required video codec.

NVC follows the following procedure to configure the media profile.

Retrieve media profiles by invoking GetProfiles command.

Retrieve supported video encoder configuration options for a media profile by invoking GetVideoEncoderConfigurationOptions (**media profile token**) command. Check whether the selected media profile supports the required video codec.

Repeat test procedure-2 for all media profiles till a media profile with the required video codec support is found.



## A.6 Media Profile Configuration for Audio & Video Streaming

For the execution of real time streaming – Audio & Video test cases, NVC has to select and configure the media profile based on the required audio & video codec.

NVC follows the following procedure to configure the media profile.

Retrieve media profiles by invoking GetProfiles command.

Retrieve supported video encoder configuration options for a media profile by invoking GetVideoEncoderConfigurationOptions (**media profile token**) command. Check whether the selected media profile supports the required video codec.

If the media profile includes audio source and audio encoder configurations

Retrieve supported audio encoder configuration options for the media profile by invoking GetAudioEncoderConfigurationOptions (**media profile token**) command. Check whether the selected media profile supports the required audio codec.

If the media profile doesn't have audio source and audio encoder configuration

Retrieve audio source configurations of DUT by invoking GetAudioSourceConfigurations command.

Add audio source configuration to the profile by invoking AddAudioSourceConfigurations command.

Retrieve audio encoder configurations of DUT by invoking GetAudioEncoderConfigurations command.

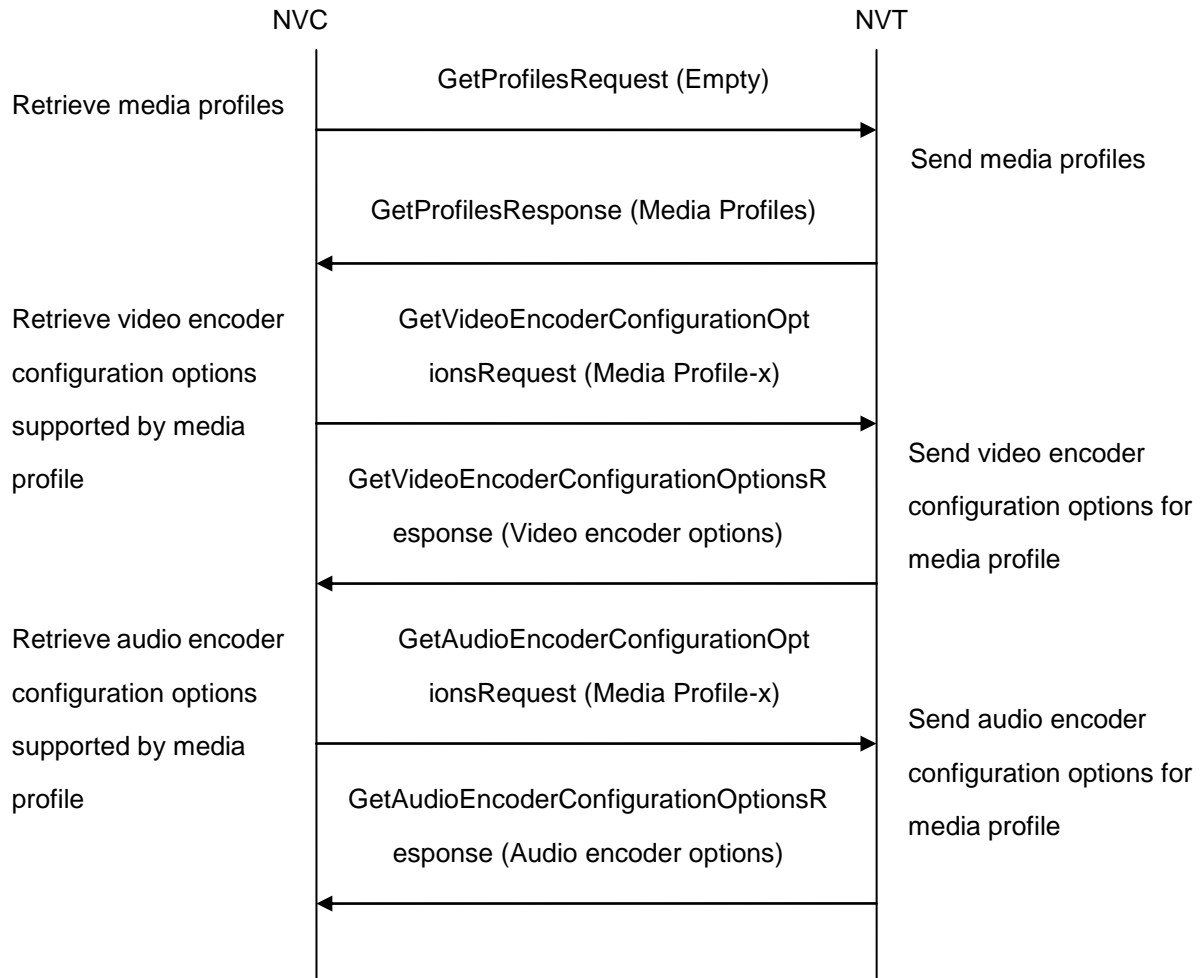
Retrieve audio encoder configuration options supported for an audio encoder configuration by invoking GetAudioEncoderConfigurationOptions (**audio encoder config token**) command. Check whether the selected audio encoder configuration supports the required audio codec.

Repeat test procedure – 4.d for all audio encoder configurations till a audio encoder configuration with the required audio codec is found.

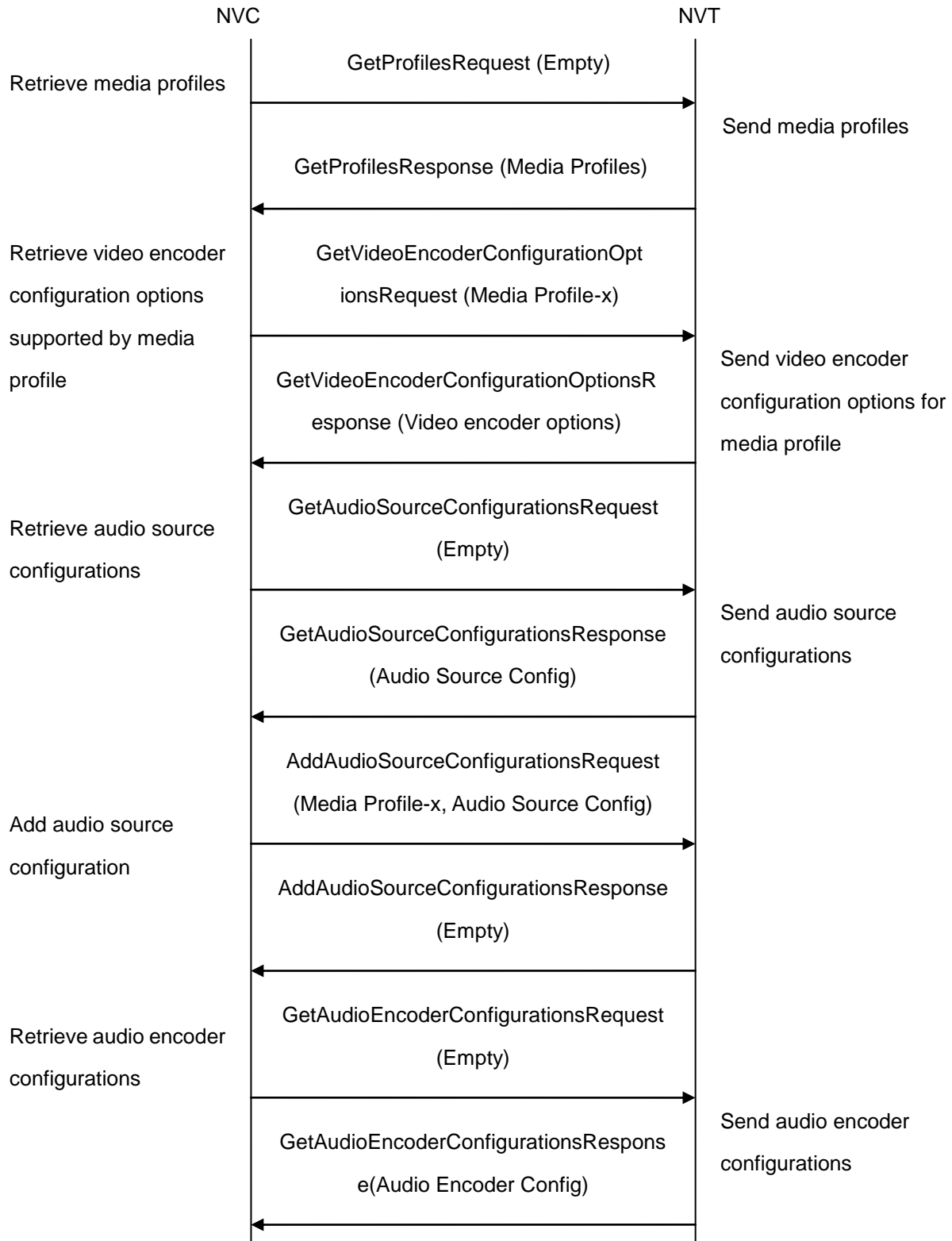
Add the corresponding audio encoder configuration to the media profile by invoking AddAudioEncoderConfiguration command.

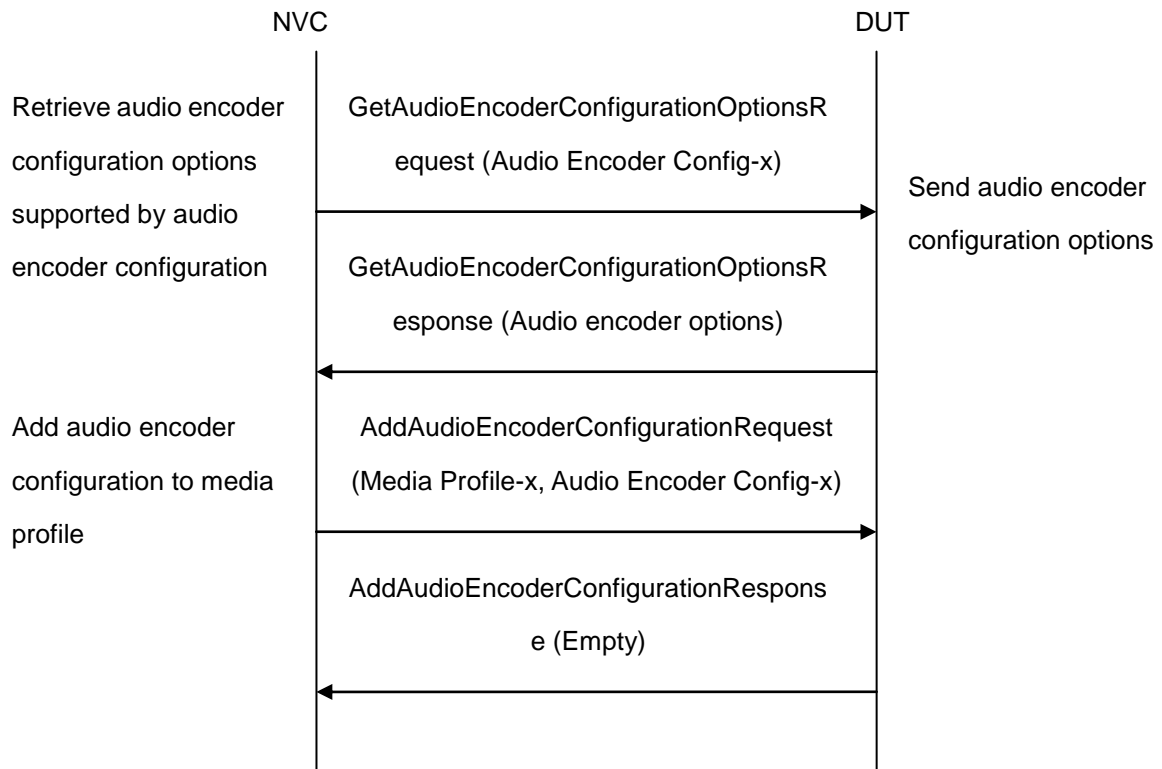
Repeat test procedure-2, 3 & 4 for all media profiles till a media profile with the required audio and video codec support is found.

Case 1: Media profile includes audio source and audio encoder configurations:



Case 2: Media profile without audio source and audio encoder configurations:







## A.7 MetadataConfiguration for receiving / not receiving events metadata

When receiving metadata a client might be interested in receiving all, none or some of the events produced by DUT. The basic rules for configuring a MetadataConfiguration to achieve this are:

To get all events: Include the Events element but do not include a filter.

To get no events: Do not include the Events element.

To get only some events: Include the Events element and include a filter in the element.

This means that if an Events element is **not** included in a MetadataConfiguration, then **no** Events metadata shall be included in the metadata stream for that configuration. Similarly, if an "<Events>" tag **without any sub-tags** is included in a MetadataConfiguration it means that **all** available Events metadata shall be included in the metadata stream for that configuration.

Example:

The following SetMetadataConfigurationRequest can be used when the client is interested in a metadata stream that includes all Events, but nothing else. The PTZStatus element is not included in the configuration, so no PTZ metadata will be included in the metadata stream. The Events element is included, but without any filter, so all Events will be included in the metadata stream.

```
<SetMetadataConfiguration>
<Configuration token="Test">
<Name>TestName</Name>
<UseCount>1</UseCount>
<Events/>
<Analytics>false</Analytics>
</Configuration>
<ForcePersistence>true</ForcePersistence>
</SetMetadataConfiguration>
```

## A.8 Multicast specific field in SDP (RTSP DESCRIBE response)

In any case that multicast real-time streaming is involved in conjunction with RTSP, **c=** field should be included to clearly indicate the multicast address being used for streaming. Following is the example of how the **c=** field should be formed in the SDP (RTSP DESCRIBE response).

```
c=IN IP4 224.10.10.100/127
```