ONVIF™

Profile S Client Test Specification

Version 16.07

July 2016

REVISION HISTORY

Vers.	Date	Description
16.07	Jun 14, 2016	Test steps sequence was changed in the following test cases: VIDEOSTREAMING-1, VIDEOSTREAMING-2, VIDEOSTREAMING-3, AUDIOSTREAMING-2, AUDIOSTREAMING-3, AUDIOSTREAMING-4.
16.07	May 27, 2016	The following test case was updated: MULTICASTSTREAMING-2 MULTICAST STREAMING USING SOAP
16.07	Apr 18, 2016	Step description in Test Procedure was updated for the test cases: MEDIASTREAMING-3, MEDIASTREAMING-4, MEDIASTREAMING-5, VIDEOSTREAMING-1, VIDEOSTREAMING-2, VIDEOSTREAMING-3, MULTICASTSTREAMING-1, AUDIOSTREAMING-2, AUDIOSTREAMING-3, AUDIOSTREAMING-4, MULTIPLEVIDEOSOURCES-1, MULTIPLEAUDIOSOURCES-1 Old description: Device response has code RTSP 200 OK if it is detected New description: If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK
16.07	Apr 15, 2016	PTZ - Continuous Positioning scenario was updated PTZCONTINUOUSPOSITIONING-3 PTZ STOP test case was replaced by two test cases: PTZCONTINUOUSPOSITIONING-3 PTZ STOP and PTZCONTINUOUSPOSITIONING-4 STOP MOVEMENT USING PTZ CONTINUOUS MOVE New Pre-Requisite added for PTZCONTINUOUSPOSITIONING-1 PTZ CONTINUOUS MOVE PAN/TILT: Device supports PTZContinuousPanTilt New Pre-Requisite added for PTZCONTINUOUSPOSITIONING-2 PTZ CONTINUOUS MOVE ZOOM: Device supports PTZContinuousZoom NOTE was removed from PTZCONTINUOUSPOSITIONING-1 PTZ CONTINUOUS MOVE PAN/TILT NOTE was removed from PTZCONTINUOUSPOSITIONING-2 PTZ CONTINUOUS MOVE ZOOM
16.07	Mar 18, 2016	Checking of TEARDOWN response was changed in Test Procedure and PASS criteria for the test cases and annexes: MEDIASTREAMING-3, MEDIASTREAMING-4, MEDIASTREAMING-5, VIDEOSTREAMING-1, VIDEOSTREAMING-2, VIDEOSTREAMING-3, MULTICASTSTREAMING-1, AUDIOSTREAMING-2, AUDIOSTREAMING-3, AUDIOSTREAMING-4, MULTIPLEVIDEOSOURCES-1, MULTIPLEAUDIOSOURCES-1, Annex A.3, Annex A.6 Old description of checking of TEARDOWN response in Test Procedure: Device responds with code RTSP 200 OK.

		New description of checking of TEARDOWN response in Test Procedure: Device response has code RTSP 200 OK if it is detected. Old description of checking of TEARDOWN response in PASS criteria: Device response on the RTSP TEARDOWN request fulfills the following requirements: New description of checking of TEARDOWN response in PASS criteria: If there is Device response on the RTSP TEARDOWN request then it fulfills the following requirements:
16.07	Mar 16, 2016	Docbook stylesheets were updated.
16.07	Mar 14, 2016	www.onvif.org was removed from Copyright section.
16.07	Mar 09, 2016	Minor changes: typos were fixed.
16.07	Feb 24, 2016	Multiple Audio Sources Test Cases were added Annex A.4 Get Audio Sources List from GetProfiles responses was added Annex A.5 Get Audio Source Token That was Used for Streaming Annex A.6 Find Audio Streaming corresponding to GetStreamUri was added
16.07	Feb 16, 2016	Multiple Video Sources Test Cases were added Annex A.1 Get Video Sources List from GetProfiles was added Annex A.2 Get Video Source Token That was Used for Streaming was added Annex A.3 Find Video Streaming corresponding to GetStreamUri was added
16.07	Feb 08, 2016	Video Source Configurations Test Cases were updated: Profile S Requirement of LIST VIDEO SOURCE CONFIGURATIONS test was changed to Optional Profile S Requirement of GET SPECIFIC VIDEO SOURCE CONFIGURATION test was changed to Optional MODIFY VIDEO SOURCE CONFIGURATION test was split to tree tests: GET VIDEO SOURCE CONFIGURATION OPTIONS, SET VIDEO SOURCE CONFIGURATION and GET COMPATIBLE VIDEO SOURCE CONFIGURATIONS. Video Encoder Configurations Test Cases were updated: Profile S Requirement of LIST VIDEO ENCODER CONFIGURATIONS test was changed to Optional Profile S Requirement of GET SPECIFIC VIDEO ENCODER CONFIGURATION test was changed to Optional MODIFY VIDEO ENCODER CONFIGURATION test was split to two tests: GET VIDEO ENCODER CONFIGURATION OPTIONS and SET VIDEO ENCODER CONFIGURATION.
16.07	Jan 26, 2016	The description about structure and hierarchy was replaced for the test cases: MEDIASTREAMING-1, MEDIASTREAMING-2, MULTICASTSTREAMING-2, VIDEOENCODERCONFIGURATIONS-1, VIDEOENCODERCONFIGURATION-2, VIDEOENCODERCONFIGURATION-3,

MEDIAPROFILECONFIGURATIONS-1, MEDIAPROFILECONFIGURATIONS-2, MEDIAPROFILECONFIGURATIONS-3, VIDEOSOURCECONFIGURATIONS-1, VIDEOSOURCECONFIGURATIONS-2, VIDEOSOURCECONFIGURATION-3, VIDEOSOURCECONFIGURATION-4, PTZLISTING-1, PTZLISTING-2, PTZCONFIGURATION-1, PTZCONFIGURATION-1, PTZCONTINUOUSPOSITIONING-1, PTZCONTINUOUSPOSITIONING-2, PTZABSOLUTEPOSITIONING-1, PTZABSOLUTEPOSITIONING-2, PTZRELATIVEPOSITIONING-1, PTZRELATIVEPOSITIONING-2, PTZPRESETS-1, PTZPRESETS-2, PTZHOMEPOSITION-1

Old description:

Client %COMMAND NAME% request message is a well-formed SOAP request (refer to onvif.xsd) AND

Client %COMMAND NAME% request message has a proper hierarchy (refer to %SERVICE%.wsdl) AND

New description:

Client %COMMAND NAME% request messages are valid according to XML Schemas listed in Namespaces AND

Client %COMMAND NAME% request in Test Procedure fulfills the following requirements:

The following steps was removed because the requirements are fullfield by XML Schemas validation:

- MEDIASTREAMING-2:

  [S2] "<GetStreamUri>" includes tag: "<StreamSetup>" AND

  [S3] "<StreamSetup>" includes tag: "<Stream>" with ("RTP-Unicast" OR "RTP-Multicast") value AND

  [S4] "<StreamSetup>" includes tag: "<Transport>" AND
- PTZCONTINUOUSPOSITIONING-1:

  [S3] "<ContinuousMove>" includes tag: "<Velocity>" AND

  [S5] "<PanTilt>" tag contains attribute: "x=" with value (example: -1, 0.1, 1, ...) AND

  [S6] "<PanTilt>" tag contains attribute: "y=" with value (example: -1, 0.1, 1, ...) AND
- PTZCONTINUOUSPOSITIONING-2:

  [S3] "<ContinuousMove>" includes tag: "<Velocity>" AND

  [S5] "<Zoom>" tag contains attribute: "x=" with value (example: -1, 0.1, 1, ...) AND
- PTZABSOLUTEPOSITIONING-1:

  [S3] "<AbsoluteMove>" includes tag: "<Position>" AND

  [S5] "<PanTilt>" tag contains attribute: "x=" with value (example: -1, 0.1, 1, ...) AND

  [S6] "<PanTilt>" tag contains attribute: "y=" with value (example: -1, 0.1, 1, ...) AND

		• PTZABSOLUTEPOSITIONING-2: [S3] "<AbsoluteMove>" includes tag: "<Position>" AND [S5] "<Zoom>" tag contains attribute: "x=" with value (example: -1, 0.1, 1, ...) AND • PTZRELATIVEPOSITIONING-1: [S3] "<RelativeMove>" includes tag: "<Translation>" AND [S5] "<PanTilt>" tag contains attribute: "x=" with value (example: -1, 0.1, 1, ...) AND [S6] "<PanTilt>" tag contains attribute: "y=" with value (example: -1, 0.1, 1, ...) AND • PTZRELATIVEPOSITIONING-2: [S3] "<RelativeMove>" includes tag: "<Translation>" AND [S5] "<Zoom>" tag contains attribute: "x=" with value (example: -1, 0.1, 1, ...) AND
16.07	Jan 11, 2016	The following test cases were updated to check of corresponding between RTSP session and GetStreamUri: MULTICAST STREAMING USING RTSP Normative references were updated.
16.07	Dec 30, 2015	The following test cases were updated to check of corresponding between RTSP session and GetStreamUri: STREAMING OVER RTSP STREAMING OVER UDP STREAMING OVER HTTP Normative references were updated.
16.01	Dec 28, 2015	The following test cases were updated to check of media type in RTSP SETUP requests and to check of corresponding between RTSP session and GetStreamUri: MJPEG VIDEO STREAMING MPEG4 VIDEO STREAMING H264 VIDEO STREAMING G.711 AUDIO STREAMING G.726 AUDIO STREAMING AAC AUDIO STREAMING Normative references were updated.
16.01	December 02, 2015	• Media Streaming Feature was updated to requiere supporting of RTP/UDP or RTP/RTSP/HTTP/TCP.
16.07	Nov 27, 2015	General item (Test Owerview) was added Minor updates in formatting, typos and terms Metadata Configurations test cases and related feature were updated according review results.
16.01	Sep 23, 2015	Added new Test Cases sections: Metadata Configurations. PTZ SEND AUXILIARY COMMAND test case was updated
15.06	Jun 10, 2015	No major changes were made, just minor formatting fixes.
15.05	May 20, 2015	No major changes were made, just minor grammatical corrections.
15.02	Feb 19, 2015	Pass criteria in VIDEOSTREAMING-1, 2 and 3 test cases have been updated (added check for Media Type: "video" in RTSP DESCRIBE response).
14.12	Dec 11, 2014	Fixed typos and inconsistencies.

1.2	Oct 29, 2014	Changes were made in "PASS" criteria of the "5.4. STREAMING OVER RTSP", "6.2. MJPEG VIDEO STREAMING", "6.3. MPEG4 VIDEO STREAMING", "6.4. H264 VIDEO STREAMING" and "7.2. MULTICAST STREAMING USING RTSP" Test Cases. Test "5.4. STREAMING OVER RTSP" was divided into three different tests (RTSP, UDP and HTTP). New Test Case "8.3. GET SPECIFIC VIDEO ENCODER CONFIGURATION" has been added. Section "10.1. Expected Scenarios Under Test" has been updated. New Test Case "10.3. GET SPECIFIC VIDEO SOURCE CONFIGURATION" has been added. "ISO/IEC Directives, Part 2" reference has been added to "2. Normative references" section. The new section "3.1 Conventions" has been added. Specific Namespace prefixes have been removed from "PASS" criteria of all Test Cases. Fixed typos and inconsistencies.
1.1	Sep 04, 2014	MEDIASTREAMING-1, MEDIASTREAMING-2 and MEDIASTREAMING-3 test cases have been updated. Video Streaming Test Cases have been added. Multicast Streaming Test Cases have been added. Test Cases for Video Encoder Configurations have been added. Media Profile Configurations Test Cases have been added. Video Source Configurations Test Cases have been added. "Scope", "Security", "Capabilities" and "Event Handling" sections have been updated.
1.0	Jul 31, 2014	Initial version. The first release includes MEDIASTREAMING-1 GET PROFILES, MEDIASTREAMING-2 GET STREAM URI and MEDIASTREAMING-3 STREAMING OVER RTSP test cases.

Table of Contents

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. This specification also acts as an input document to the development of a test tool which will be used to test the ONVIF Client implementation conformance towards ONVIF standard. This Client Test Tool analyzes network communications between ONVIF Devices and Clients being tested and determines whether a specific Client is ONVIF conformant (see ONVIF Conformance Process Specification).

This particular document defines test cases required for testing Profile S features of a Client application e.g. Media Streaming, Video Streaming, Multicast Streaming, Video Encoder Configuration, Media Profile Creation and Video Source Configuration. It also describes the test framework, test setup, prerequisites, test policies needed for the execution of the described test cases.

1.1 Scope

This ONVIF Profile S Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile S features. Conformance testing is meant to be black-box network traces analysis and verification. The objective of this specification is to provide the test cases to test individual requirements of ONVIF Clients in the scope of Profile S features according to ONVIF Profile Specifications.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Profile S features.

This specification **does not** address the following:

- Product use cases and non-functional (performance and regression) testing and analysis.
- SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
- Network protocol implementation Conformance test for HTTPS, HTTP, RTP and RTSP protocols.

The following sections cover test cases needed for the verification of relevant features as mentioned in the ONVIF Profile Specifications.

1.2 Media Streaming

Media Streaming section defines different streaming options based on RTP protocol which are required for all types of streams of video, audio and metadata. Media control is done using RTSP protocol.

1.3 Video Streaming

Video Streaming section specifies Client ability to establish specific video streams in MJPEG, MPEG4 and H264 video formats.

1.4 Multicast Streaming

Multicast Streaming section specifies Client ability to initiate multicast stream by using StartMulticastStreaming and StopMulticastStreaming operations or by using RTSP SETUP command with multicast transport parameter.

1.5 Video Encoder Configuration

Video Encoder Configurations section specifies listing and modification of video encoder configurations on Device.

1.6 Media Profile Configurations

Media Profile Configurations section specifies creation and retrieval of Media Profile Configurations from Device.

1.7 Video Source Configuration

Video Source Configurations section specifies listing and modification of video source configurations on Device.

1.8 PTZ - Listing

PTZ - Listing section specifies Client ability to read PTZ capabilities.

1.9 PTZ - Configuration

PTZ - Configuration section specifies Client ability to add PTZ configuration to a media profile.

1.10 PTZ - Continuous Positioning

PTZ - Continuous Move section specifies Client ability to move a PTZ Device using ContinuousMove operation and stop ongoing movement using Stop operation or sending zero values for Pan/Tilt and Zoom.

1.11 PTZ - Absolute Positioning

PTZ - Absolute Positioning section specifies Client ability to move a PTZ Device using the AbsoluteMove operation.

1.12 PTZ - Relative Positioning

PTZ - Relative Positioning section specifies Client ability to move a PTZ Device using the RelativeMove operation.

1.13 PTZ - Presets

PTZ - Presets section specifies Client ability to list the presets of a PTZ Node and move a PTZ Device to a specific preset.

1.14 PTZ - Home Position

PTZ - Home Position section specifies Client ability to move a PTZ Device to its home position.

1.15 PTZ - Auxiliary Command

PTZ - Auxiliary Command section specifies Client ability to send auxiliary commands to a PTZ Device.

1.16 Audio Streaming

Audio Streaming section specifies Client ability to initiate audio stream in G.711, G.726 and AAC encoding formats. This section also specifies Client ability to configure a media profile for audio streaming.

1.17 Metadata Configuration

Metadata Configurations section specifies listing and modification of metadata configurations on Device.

1.18 Multiple Video Sources

Multiple Video Sources section specifies Client ability to initiate video streaming for all Video Sources returned by Device in GetProfilesResponse.

1.19 Multiple Audio Sources

Multiple Audio Sources section specifies Client ability to initiate audio streaming for all Audio Sources returned by Device in GetProfilesResponse.

2 Normative references

- ONVIF Conformance Process Specification:

  http://www.onvif.org/Documents/Specifications.aspx

- ONVIF Profile Policy:

  http://www.onvif.org/Documents/Specifications.aspx

- ONVIF Core Specifications:

  http://www.onvif.org/Documents/Specifications.aspx

- ONVIF Core Client Test Specification:

  http://www.onvif.org/Documents/Specifications.aspx

- ONVIF Profile S Specification version 1.0, Dec 2011:

  http://www.onvif.org/Documents/Specifications.aspx

- ONVIF Streaming Specification:

  http://www.onvif.org/Documents/Specifications.aspx

- ONVIF Media Service Specification:

  http://www.onvif.org/Documents/Specifications.aspx

- ISO/IEC Directives, Part 2, Annex H:

  www.iso.org/directives

- ISO 16484-5:2014-09 Annex P:

  https://www.iso.org/obp/ui/#!iso:std:63753:en

- WS-BaseNotification:

  http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

- W3C SOAP 1.2, Part 1, Messaging Framework:

  http://www.w3.org/TR/soap12-part1/

- W3C XML Schema Part 1: Structures Second Edition:

  http://www.w3.org/TR/xmlschema-1/

- W3C XML Schema Part 2: Datatypes Second Edition:

  "http://www.w3.org/TR/xmlschema-2/ [http://www.w3.org/TR/xmlschema-2/]
- W3C Web Services Addressing 1.0 – Core:

  http://www.w3.org/TR/ws-addr-core/
- IETF RFC 2435, RTP Payload Format for JPEG-compressed Video:

  http://www.ietf.org/rfc/rfc2435.txt
- IETF RFC 3016, RTP Payload Format for MPEG-4 Audio/Visual Streams:

  http://www.ietf.org/rfc/rfc3016
- IETF RFC 3984, RTP Payload Format for H.264 Video:

  http://www.ietf.org/rfc/rfc3984
- IETF RFC 3640, RTP Payload Format for Transport of MPEG-4 Elementary Streams:

  http://www.ietf.org/rfc/rfc3640
- IETF RFC 2326, Real Time Streaming Protocol (RTSP):

  http://www.ietf.org/rfc/rfc2326.txt

3 Terms and Definitions

3.1 Conventions

The key words "shall", "shall not", "should", "should not", "may", "need not", "can", "cannot" in this specification are to be interpreted as described in [ISO/IEC Directives Part 2].

3.2 Definitions

This section describes terms and definitions used in this document.

Address	An address refers to a URI.
Profile	See ONVIF Profile Policy.
Profile S	The Profile S Specification.
ONVIF Device	Computer appliance or software program that exposes one or multiple ONVIF Web Services.
ONVIF Client	Computer appliance or software program that uses ONVIF Web Services.
Conversation	A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.
Network	A network is an interconnected group of devices communicating using the Internet protocol.
Network Trace Capture file	Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.
Media Profile	A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations.
SOAP	SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.
Client Test Tool	ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.
Valid Device Response	Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.
PTZ Node	Low-level PTZ entity that maps to the PTZ Device and its capabilities.

3.3 Abbreviations

This section describes abbreviations used in this document.

JPEG Joint Photographic Expert Group.

MPEG-4 Moving Picture Experts Group 4.

HTTP Hyper Text Transport Protocol.

HTTPS Hyper Text Transport Protocol over Secure Socket Layer.

RTP Real-time Transport Protocol.

RTSP Real Time Streaming Protocol.

SDP Session Description Protocol.

TCP Transport Control Protocol.

UDP User Datagram Protocol.

URI Uniform Resource Identifier.

WSDL Web Services Description Language.

XML eXtensible Markup Language.

PTZ Pan/Tilt/Zoom.

3.4 Namespaces

Prefix and namespaces used in this test specification are listed in Table 1. These prefixes are not part of the standard and an implementation can use any prefix.

Table 3.1. Defined namespaces in this specification

Prefix	Namespace URI	Description
soapenv	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]
xs	http://www.w3.org/2001/XMLSchema	Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2]
xsi	http://www.w3.org/2001/XMLSchema-instance	XML schema instance namespace
tns1	http://www.onvif.org/ver10/topics	The namespace for the ONVIF topic namespace
tt	http://www.onvif.org/ver10/schema	ONVIF XML schema descriptions
tds	http://www.onvif.org/ver10/device/wsdl	The namespace for the WSDL device service
trt	http://www.onvif.org/ver10/media/wsdl	The namespace for the WSDL media service
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
tptz	http://www.onvif.org/ver20/ptz/wsdl	The namespace for the WSDL PTZ service
wsnt	http://docs.oasis-open.org/wsn/b-2	Schema namespace of the [WS-BaseNotification] specification.

Prefix	Namespace URI	Description
wsa	http://www.w3.org/2005/08/addressing	Device addressing namespace as defined by [WS-Addressing].

4 Test Overview

This section provides information for the test setup procedure and required prerequisites that should be followed during test case execution.

An ONVIF Client compliant to the Profile S is an ONVIF Client that can configure, request, and control streaming of video data over an IP network from an ONVIF Device compliant to the Profile S. The Profile S also includes support for control of PTZ, receiving Audio and Metadata Stream, and Relay Outputs.

An ONVIF Profile is described by a fixed set of functionalities through a number of services that are provided by the ONVIF standard. A number of services and functionalities are mandatory for each type of ONVIF Profile. An ONVIF Device and ONVIF Client may support any combination of Profiles and other optional services and functionalities.

4.1 General

Test Cases are grouped depending on features. Each Test Cases group provides description of feature requirement level for Profiles, expected scenario under test and related test cases:

- Feature Level Requirement
- Expected Scenarios Under Test
- List of Test Cases

4.1.1 Feature Level Requirement

Feature Level Requirement item contains a feature ID and feature requirement level for the Profiles, which will be used for Profiles conformance.

If Feature Level Requirement is defined as Mandatory for some Profile, Client shall pass Expected Scenario Under Test for each Device with this Profile support to claim this Profile Conformance.

If Feature Level Requirement is defined as Conditional, Optional for some Profile, Client shall pass Expected Scenario Under Test for at least one Device with this Profile support to claim feature as supported.

4.1.2 Expected Scenarios Under Test

Expected Scenarios Under Test item contains expected scenario under test, conditions when the feature will be defined as supported and as not supported.

4.1.3 Test Cases

Test Case items contain list of test cases which are related to feature. Test cases provide exact procedure of testing feature support conditions.

Each Test Case contains the following parts:

- Test Label - Unique label for each test
- Test Case ID - Unique ID for each test
- Profile Normative References - Requirement level for the feature under test is defined in Profile Specification. This reference is informative and will not be used in conformance procedure.
- Feature Under Test - Feature which is under current test. Typically a particular command or an event.
- Test Purpose - The purpose of current test case.
- Pre-Requisite - The pre-requisite defines when the test should be performed. In case if pre-reqiusite does not match, the test result will be NOT DETECTED.
- Test Procedure - scenario expected to be reflected in network trace file.
- Test Result - Passed and failed criteria of the test case. Depending on these criteria test result will be defined as PASSED or FAILED.
- Validated Feature List - list of features ID related to this test case.

4.2 Test Setup

Collect Network traces files required by the test cases.

Collect Feature List XML files for Devices detected in the Network Trace files.

Client shall support all mandatory and conditional features listed in the Device Feature List XML file supplied for the Profiles supported by the Client.

For compatibility with the Profile S, the ONVIF Client shall follow the requirements of the conformance process. For details please see the latest ONVIF Conformance Process Specification.

4.3 Prerequisites

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

The Device shall be configured with an IPv4 address.

The Device shall be able to be discovered by the Client.

5 Media Streaming Test Cases

5.1 Feature Level Requirement:

Validated Feature: MediaStreaming

Profile S Requirement: Mandatory

5.2 Expected Scenarios Under Test:

1. Client connects to Device to initiate Media Streaming.
2. Client is considered as supporting Media Streaming if the following conditions are met:
   - Device returns a valid response to GetProfiles request AND
   - Device returns a valid response to GetStreamURI request AND
   - Stream was successfully established by Client using UDP protocol OR HTTP protocol.
   - Stream was successfully established by Client using RTSP protocol (if supported).
3. Client is considered as NOT supporting Media Streaming if the following is TRUE:
   - No Valid Device Response to GetProfiles request OR
   - No Valid Device Response to GetStreamURI request OR
   - Client is unable to establish stream using UDP protocol OR HTTP protocol OR
   - Client is unable to establish stream using RTSP protocol if detected.

5.3 GET PROFILES

Test Label: Media Streaming - GetProfiles

Test Case ID: MEDIASTREAMING-1

Profile S Normative Reference: Mandatory

Feature Under Test: Media Streaming

Test Purpose: To verify that list of media profiles from Device is received by Client using the GetProfiles operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetProfiles operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetProfiles request message to retrieve complete profiles list from Device.
2. Device responds with code HTTP 200 OK and GetProfilesResponse message.

Test Result:

PASS -

- Client **GetProfiles** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetProfiles** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetProfiles>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetProfilesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaStreaming_GetProfiles

5.4 GET STREAM URI

Test Label: Media Streaming - GetStreamURI

Test Case ID: MEDIASTREAMING-2

Profile S Normative Reference: Mandatory

Feature Under Test: Media Streaming

Test Purpose: To verify that stream URI from Device is received by Client using the GetStreamURI operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetStreamURI operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetStreamUri request message with the Stream Setup element (contains two parts: Stream Type and Transport protocol) and Profile Token element (indicates the media profile selected).
2. Device responds with code HTTP 200 OK and GetStreamUriResponse message.

Test Result:

PASS -

- Client **GetStreamUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetStreamUri>" tag after the "<Body>" tag AND
  - [S5] "<Transport>" includes tag: "<Protocol>" with ("UDP" OR "HTTP" OR "RTSP") value AND
  - [S6] "<GetStreamUri>" includes tag: "<ProfileToken>" with non-empty string value of specific profile token AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<GetStreamUriResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaStreaming_GetStreamURI

5.5 STREAMING OVER RTSP

Test Label: Media Streaming - RTSP

Test Case ID: MEDIASTREAMING-3

Profile S Normative Reference: governed by business rule #21

Feature Under Test: Media Streaming

Test Purpose: To verify that stream over RTSP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/TCP" and which does not contain Require header with "onvif-replay" value and which is not tunneled in HTTP present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile with Stream Type element with "RTP-Unicast" OR "RTP-Multicast" value and Transport Protocol element with "RTSP" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message. message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request with **Transport** tag in RTSP header that contains "RTP/AVP/TCP" and without "onvif-replay" Require header to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/TCP" (transport=RTP, profile=AVP, lower-transport=TCP) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
  - [S3] It is not tunneled in HTTP AND

- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S4] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S6] It invoked before the Client **RTSP SETUP** request AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
  - [S8] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S9] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S11] **trt:StreamSetup/tt:Transport/tt:Protocol** element value is equal to "RTSP"
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S12] It has HTTP 200 response code AND
  - [S13] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S14] It is invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S15] It invoked after the Client **RTSP SETUP** request AND
  - [S16] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S17] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S18] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S19] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S20] It invoked after the Client **RTSP PLAY** request AND
  - [S21] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S22] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaStreaming_RTPRTSPTCP

5.6 STREAMING OVER UDP

Test Label: Media Streaming - UDP

Test Case ID: MEDIASTREAMING-4

Profile S Normative Reference: governed by business rule #21

Feature Under Test: Media Streaming

Test Purpose: To verify that stream over UDP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/UDP" or "RTP/AVP" and which does not contain Require header with "onvif-replay" value present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile with Stream Type element with "RTP-Unicast" OR "RTP-Multicast" value and Transport Protocol element with "UDP" value.

2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message. message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request with **Transport** tag in RTSP header that contains "RTP/AVP/UDP" or "RTP/AVP" and without "onvif-replay" Require header to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/UDP" OR "RTP/AVP" (transport=RTP, profile=AVP, lower-transport=TCP or skipped) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S3] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S5] It invoked before the Client **RTSP SETUP** request AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:

- [S6] It has RTSP 200 response code AND
- [S7] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND

- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S9] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S10] **trt:StreamSetup/tt:Transport/tt:Protocol** element value is equal to "UDP"
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S11] It has HTTP 200 response code AND
  - [S12] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S13] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S14] It invoked after the Client **RTSP SETUP** request AND
  - [S15] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S17] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S18] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S19] It invoked after the Client **RTSP PLAY** request AND
  - [S20] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaStreaming_RTPUDP

5.7 STREAMING OVER HTTP

Test Label: Media Streaming - HTTP

Test Case ID: MEDIASTREAMING-5

Profile S Normative Reference: governed by business rule #21

Feature Under Test: Media Streaming

Test Purpose: To verify that stream over HTTP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/TCP" and which does not contain Require header with "onvif-replay" value and which is tunneled in HTTP present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile with Stream Type element with "RTP-Unicast" OR "RTP-Multicast" value and Transport Protocol element with "RTSP" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message. message.
3. Client invokes **RTSP DESCRIBE** request in HTTP tunnel to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header in HTTP tunnel with **Transport** tag in RTSP header that contains "RTP/AVP/TCP" to set media session parameters.

6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header in HTTP tunnel to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request in HTTP tunnel to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/TCP" (transport=RTP, profile=AVP, lower-transport=TCP) (see [RFC 2326]) AND
  - [S2] It is tunneled in HTTP AND
  - [S3] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S4] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S6] It invoked before the Client **RTSP SETUP** request AND
  - [S7] It is tunneled in HTTP AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S8] It has RTSP 200 response code AND
  - [S9] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S10] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It invoked before the Client **RTSP DESCRIBE** request AND

- [S12] **trt:StreamSetup/tt:Transport/tt:Protocol** element value is equal to "HTTP"

- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S13] It has HTTP 200 response code AND
  - [S14] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S15] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S16] It invoked after the Client **RTSP SETUP** request AND
  - [S17] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S18] It is tunneled in HTTP AND
  - [S19] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S20] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S21] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S22] It invoked after the Client **RTSP PLAY** request AND
  - [S23] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S24] It is tunneled in HTTP AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S25] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaStreaming_RTPRTSPHTTP

6 Video Streaming Test Cases

6.1 Feature Level Requirement:

Validated Feature: VideoStreaming

Profile S Requirement: Mandatory

6.2 Expected Scenarios Under Test:

1. Client connects to Device to initiate Video Streaming of a specific encoding type.
2. Client is considered as supporting Video Streaming if the following conditions are met:
   - Client is able to initiate and retrieve a video stream with MJPEG encoding type (when the device doesn't support optional encoding features) OR
   - Client is able to initiate and retrieve a video stream with MJPEG encoding AND support all optional encodings (when the device supports optional encodings).
3. Client is considered as NOT supporting Video Streaming if ANY of the following is TRUE:
   - MJPEG Video Streaming attempts detected have failed OR
   - (when the device supports optional MPEG4 or H264 encodings) EITHER MPEG4 Video Streaming attempts detected have failed OR H264 Video Streaming attempts detected have failed.

6.3 MJPEG VIDEO STREAMING

Test Label: Video Streaming - MJPEG

Test Case ID: VIDEOSTREAMING-1

Profile S Normative Reference: Mandatory

Feature Under Test: Video Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with MJPEG encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of MJPEG encoding type.

- Device supports JPEG encoding for Video Streaming.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with JPEG Encoding value. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "JPEG" or with payload type number "26".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for JPEG video streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] IF SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" THEN encoding name is "JPEG"
  - [S3] ELSE IF SDP packet contains media type "video" (m=video) without sessions attribute "rtpmap" THEN payload type number is "26" (see [RFC 2435]) AND

- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S5] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S6] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S7] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S8] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S9] It has HTTP 200 response code AND
  - [S10] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It received before the Client **RTSP DESCRIBE** request AND
  - [S12] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S13] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S14] It invoked after the Client **RTSP SETUP** request AND
  - [S15] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S17] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:

- [S18] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
- [S19] It invoked after the Client **RTSP PLAY** request AND
- [S20] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_streaming.jpeg

6.4 MPEG4 VIDEO STREAMING

Test Label: Video Streaming - MPEG4

Test Case ID: VIDEOSTREAMING-2

Profile S Normative Reference: Conditional

Feature Under Test: Video Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with MPEG4 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of MPEG4 encoding type.
- Device supports MPEG4 encoding for Video Streaming.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with MPEG4 Encoding value. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.

2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "MP4V-ES".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for MPEG4 video streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" with encoding name "MP4V-ES" (see [RFC 3016], item 5.2 SDP usage of MPEG-4 Visual) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked the same RTSP session as the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_streaming.mpeg4

6.5 H264 VIDEO STREAMING

Test Label: Video Streaming - H264

Test Case ID: VIDEOSTREAMING-3

Profile S Normative Reference: Conditional

Feature Under Test: Video Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with H264 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of H264 encoding type.
- Device supports H264 encoding for Video Streaming.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with H264 Encoding value. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "H264".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for H264 video streaming.

6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" with encoding name "H264" (see [RFC 3984], item 8.2.1. Mapping of MIME Parameters to SDP) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND

- [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
- [S10] It received before the Client **RTSP DESCRIBE** request AND
- [S11] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND

- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_streaming.h264

7 Multicast Streaming Test Cases

7.1 Expected Scenarios Under Test:

1. Client connects to Device and initiates Multicast Streaming using RTSP or using StartMultiCastStreaming and StopMultiCastStreaming operations.
2. Client is considered as supporting Multicast Streaming if the following conditions are met:
   - Able to start and stop a multicast stream by using Start/StopMulticastStreaming OR
   - Able to start and stop a multicast stream by using RTSP
3. Client is considered as NOT supporting Multicast Streaming if ANY of the following is TRUE:
   - If using Start/StopMulticastStreaming -> session never passed the PLAY state or was never terminated AND
   - If using RTSP -> RTSP session never passed the PLAY state or was never terminated

7.2 MULTICAST STREAMING USING RTSP

Test Label: Multicast Streaming - RTSP multicast setup

Test Case ID: MULTICASTSTREAMING-1

Profile S Normative Reference: Conditional

Feature Under Test: Multicast Streaming

Test Purpose: To verify that the Client is able to setup and initiate a multicast stream with RTSP commands for stream control.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/UDP;multicast" or "RTP/AVP;multicast" and without "onvif-replay" Require header present.
- Device supports RTPMulticastUDP feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile with Stream Type element with "RTP-Multicast" value and Transport Protocol element with "UDP" value.

2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message. message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with **Transport** tag in RTSP header that contains "RTP/AVP/UDP;multicast" or "RTP/AVP;multicast" to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/UDP" OR "RTP/AVP" and with "multicast" parameter value (transport=RTP, profile=AVP, lower-transport=TCP or skipped, parameter=multicast) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S3] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure that fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S5] It invoked before the Client **RTSP SETUP** request AND
  - [S6] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND

- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device **GetStreamUri** request in Test Procedure that fulfills the following requirements:
  - [S8] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S9] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S10] **trt:StreamSetup/tt:Stream** element value is equal to "RTP-Multicast"
  - [S11] **trt:StreamSetup/tt:Transport/tt:Protocol** element value is equal to "UDP"
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S12] It has HTTP 200 response code AND
  - [S13] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure that fulfills the following requirements:
  - [S14] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S15] It invoked after the Client **RTSP SETUP** request AND
  - [S16] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S17] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S18] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure that fulfills the following requirements:
  - [S19] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S20] It invoked after the Client **RTSP PLAY** request AND

- [S21] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:

  - [S22] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MulticastStreaming_RTSP

7.3 MULTICAST STREAMING USING SOAP

Test Label: Multicast Streaming - SOAP multicast setup

Test Case ID: MULTICASTSTREAMING-2

Profile S Normative Reference: Conditional

Feature Under Test: Multicast Streaming

Test Purpose: To verify that the Client is able to setup and initiate a multicast stream with Start/StopMulticastStreaming SOAP operations for stream control.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with StartMulticastStreaming operation for stream control.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes StartMulticastStreaming request message with non-empty ProfileToken element.
2. Device responds with code HTTP 200 OK and StartMulticastStreamingResponse message.
3. Client invokes StopMulticastStreaming request message with non-empty ProfileToken element.
4. Device responds with code HTTP 200 OK and StopMulticastStreamingResponse message.

Test Result:

NOTE: In case when StopMulticastStreaming command is detected and StartMulticastStreaming command is not detected then the test shall be deemed as "NOT DETECTED".

PASS -

- Client **StartMulticastStreaming** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **StartMulticastStreaming** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:StartMulticastStreaming** AND
  - [S2] **trt:StartMulticastStreaming/trt:ProfileToken** element has non-empty string value of specified profile token AND
- Device response on the **StartMulticastStreaming** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:StartMulticastStreamingResponse**.
- Client **StopMulticastStreaming** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **StopMulticastStreaming** request in Test Procedure fulfills the following requirements:
  - [S5] **soapenv:Body** element has child element **trt:StopMulticastStreaming** AND
  - [S6] **trt:StopMulticastStreaming/trt:ProfileToken** element has non-empty string value of specified profile token AND
- Device response on the **StopMulticastStreaming** request fulfills the following requirements:
  - [S7] It has HTTP 200 response code AND
  - [S8] **soapenv:Body** element has child element **trt:StopMulticastStreamingResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MulticastStreaming_SOAP

8 Video Encoder Configurations Test Cases

8.1 Feature Level Requirement:

Validated Feature: video_encoder_configuration

Profile S Requirement: Mandatory

8.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Video Encoder Configurations.
2. Client is considered as supporting Video Encoder Configurations if the following conditions are met:
   - Device returns a valid response to **GetVideoEncoderConfigurations** operations AND
   - Device returns a valid response to **GetVideoEncoderConfiguration** operations AND
   - Client is able to retrieve video encoder configuration options using **GetVideoEncoderConfigurationOptions** operation AND
   - Client is able to change video encoder configuration settings using **SetVideoEncoderConfiguration** operation.
3. Client is considered as NOT supporting Video Encoder Configurations if ANY of the following is TRUE:
   - No Valid Device Response to **GetVideoEncoderConfigurations** request if detected OR
   - No Valid Device Response to **GetVideoEncoderConfiguration** request if detected OR
   - No valid responses for **GetVideoEncoderConfigurationOptions** request OR
   - No valid responses for **SetVideoEncoderConfiguration** request.

8.3 LIST VIDEO ENCODER CONFIGURATIONS

Test Label: Video Encoder Configurations - list all existing video encoder configurations

Test Case ID: VIDEOENCODERCONFIGURATIONS-1

Profile S Normative Reference: Optional

Feature Under Test: Video Encoder Configurations

Test Purpose: To verify that list of all existing video encoder configurations from Device is received by Client using the GetVideoEncoderConfigurations operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetVideoEncoderConfigurations operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetVideoEncoderConfigurations request message to retrieve complete list of available video encoder configurations from Device.
2. Device responds with code HTTP 200 OK and GetVideoEncoderConfigurationsResponse message.

Test Result:

PASS -

- Client **GetVideoEncoderConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoEncoderConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetVideoEncoderConfigurations>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetVideoEncoderConfigurationsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: VideoEncoderConfigurations_GetVideoEncoderConfigurations

8.4 GET SPECIFIC VIDEO ENCODER CONFIGURATION

Test Label: Video Encoder Configurations - gets a specific encoder configuration

Test Case ID: VIDEOENCODERCONFIGURATION-2

Profile S Normative Reference: Optional

Feature Under Test: Video Encoder Configurations

Test Purpose: To verify that Client is able to retrieve a specific encoder configuration from Device by using the GetVideoEncoderConfiguration operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetVideoEncoderConfiguration operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetVideoEncoderConfiguration request message with specified ConfigurationToken.
2. Device responds with code HTTP 200 OK and GetVideoEncoderConfigurationResponse.

Test Result:

PASS -

- Client **GetVideoEncoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoEncoderConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetVideoEncoderConfiguration>" tag after the "<Body>" tag AND
  - [S2] "<GetVideoEncoderConfiguration>" includes tag: "<ConfigurationToken>" with non-empty string value of "Token=*" parameter AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetVideoEncoderConfigurationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: VideoEncoderConfigurations_GetVideoEncoderConfiguration

8.5 GET VIDEO ENCODER CONFIGURATION OPTIONS

Test Label: Video Encoder Configuration - Get Video Encoder Configuration Options

Test Case ID: VIDEOENCODERCONFIGURATIONS-3

Profile S Normative Reference: Mandatory

Feature Under Test: Get Video Encoder Configuration Options

Test Purpose: To verify that Client is able to get video encoder configuration options provided by Device using the **GetVideoEncoderConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoEncoderConfigurationOptions** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetVideoEncoderConfigurationOptions** request message to retrieve video encoder configuration options for the Device.
2. Device responds with code HTTP 200 OK and **GetVideoEncoderConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetVideoEncoderConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoEncoderConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetVideoEncoderConfigurationOptions** AND
  - [S2] If it contains **trt:ConfigurationToken** element THEN it has non-empty string value AND
  - [S3] If it contains **trt:ProfileToken** element THEN it has non-empty string value AND
- Device response to the **GetVideoEncoderConfigurationOptions** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **trt:GetVideoEncoderConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_encoder_configuration.get_video_encoder_configuration_options

8.6 SET VIDEO ENCODER CONFIGURATION

Test Label: Configure Video Encoder Configuration - Set Video Encoder Configuration

Test Case ID: VIDEOENCODERCONFIGURATIONS-4

Profile S Normative Reference: Mandatory

Feature Under Test: Set Video Encoder Configuration

Test Purpose: To verify that Client is able to change video encoder configuration provided by Device using the **SetVideoEncoderConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetVideoEncoderConfiguration** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetVideoEncoderConfiguration** request message to change video encoder configuration on the Device.
2. Device responds with code HTTP 200 OK and **SetVideoEncoderConfigurationResponse** message.

Test Result:

PASS -

- Client **SetVideoEncoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetVideoEncoderConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:SetVideoEncoderConfiguration** AND
  - [S2] **trt:SetVideoEncoderConfiguration/trt:Configuration/@token** element has non-empty string value AND

- Device response to the **SetVideoEncoderConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **trt:SetVideoEncoderConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_encoder_configuration.set_video_encoder_configuration

9 Media Profile Configurations Test Cases

9.1 Expected Scenarios Under Test:

1. Client connects to Device to retrieve and/or create Media Profile Configuration.
2. Client is considered as supporting Media Profile Configuration if the following conditions are met:
   - Client shall be able to list available profiles using EITHER GetProfiles OR GetProfile operations AND
   - Client shall be able to create a media profile using the CreateProfile operation.
3. Client is considered as NOT supporting Media Profile Configuration if ANY of the following is TRUE:
   - No Valid Device Response to GetProfiles request OR
   - No Valid Device Response to GetProfile request OR
   - No Valid Device Response to CreateProfile request (except soap fault: maximumnumberofprofiles).

9.2 LIST AVAILABLE MEDIA PROFILES

Test Label: Media Profile Configurations - list available profiles

Test Case ID: MEDIAPROFILECONFIGURATIONS-1

Profile S Normative Reference: Conditional

Feature Under Test: Media Profile Configurations

Test Purpose: To verify that list of media profiles from Device is received by Client using the GetProfiles operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetProfiles operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetProfiles request message to retrieve complete profiles list from Device.

2. Device responds with code HTTP 200 OK and GetProfilesResponse message.

Test Result:

PASS -

- Client **GetProfiles** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetProfiles** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetProfiles>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetProfilesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaProfileConfigurations_GetProfiles

9.3 GET SPECIFIC MEDIA PROFILE

Test Label: Media Profile Configurations - gets a specific media profile.

Test Case ID: MEDIAPROFILECONFIGURATIONS-2

Profile S Normative Reference: Conditional

Feature Under Test: Media Profile Configurations

Test Purpose: To verify that Client is able to retrieve a specific media profile from Device by using the GetProfile operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetProfile operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetProfile request message to retrieve a specific media profile from Device.
2. Device responds with code HTTP 200 OK and GetProfileResponse message.

Test Result:

PASS -

- Client **GetProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetProfile** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetProfile>" tag after the "<Body>" tag AND
  - [S2] "<GetProfile>" includes tag: "<ProfileToken>" with non-empty string value of specific profile token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetProfileResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaProfileConfigurations_GetProfile

9.4 CREATE A MEDIA PROFILE

Test Label: Media Profile Configurations - create a media profile

Test Case ID: MEDIAPROFILECONFIGURATIONS-3

Profile S Normative Reference: Conditional

Feature Under Test: Media Profile Configurations

Test Purpose: To verify that Client is able to create a new media profile on Device by using the CreateProfile operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with CreateProfile operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes CreateProfile request message to create a new empty profile structure with no configuration entities.
2. Device responds with code HTTP 200 OK and CreateProfileResponse message.

Test Result:

PASS -

- Client **CreateProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateProfile** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<CreateProfile>" tag after the "<Body>" tag AND
  - [S2] "<CreateProfile>" includes tag: "<Name>" with non-empty string value of profile name AND
  - [S3] (Client request does not contain "<Token>" tag OR "<CreateProfile>" includes tag: "<Token>" with non-empty string value of specific profile token) AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<CreateProfileResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaProfileConfigurations_CreateProfile

10 Video Source Configurations Test Cases

10.1 Feature Level Requirement:

Validated Feature: video_source_configuration

Profile S Requirement: Conditional

10.2 Expected Scenarios Under Test:

1. Client connects to Device to list, modify and add Video Source Configurations.
2. Client is considered as supporting Video Source Configurations if the following conditions are met:
   - Device returns a valid response to **GetVideoSourceConfigurations** operations AND
   - Device returns a valid response to **GetVideoSourceConfiguration** operations AND
   - Client is able to retrieve video source configuration options using **GetVideoSourceConfigurationOptions** operation AND
   - Client is able to change video source configuration settings using **SetVideoSourceConfiguration** operation AND
   - Client is able to retrieve list of video source configurations that are compatible with a certain media profile using **GetCompatibleVideoSourceConfigurations** operation AND
   - Client is able to add video source configurations using **AddVideoSourceConfiguration** operation.
3. Client is considered as NOT supporting Video Source Configurations if ANY of the following is TRUE:
   - No Valid Device Response to **GetVideoSourceConfigurations** request if detected OR
   - No Valid Device Response to **GetVideoSourceConfiguration** request if detected OR
   - No valid responses for **GetVideoSourceConfigurationOptions** request OR
   - No valid responses for **SetVideoSourceConfiguration** request OR
   - No valid responses for **GetCompatibleVideoSourceConfigurations** request OR
   - No valid responses for **AddVideoSourceConfiguration** request.

10.3 LIST VIDEO SOURCE CONFIGURATIONS

Test Label: Video Source Configurations - list available video source configurations

Test Case ID: VIDEOSOURCECONFIGURATIONS-1

Profile S Normative Reference: Optional

Feature Under Test: Video Source Configurations

Test Purpose: To verify that list of all existing video source configurations from Device is received by Client using the GetVideoSourceConfigurations operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetVideoSourceConfigurations operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetVideoSourceConfigurations request message to retrieve complete list of available video encoder configurations from Device.
2. Device responds with code HTTP 200 OK and GetVideoSourceConfigurationsResponse message.

Test Result:

PASS -

- Client **GetVideoSourceConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoSourceConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetVideoSourceConfigurations>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetVideoSourceConfigurationsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: VideoSourceConfigurations_GetVideoSourceConfigurations

10.4 GET SPECIFIC VIDEO SOURCE CONFIGURATION

Test Label: Video Source Configurations - gets a specific video source configuration

Test Case ID: VIDEOSOURCECONFIGURATIONS-2

Profile S Normative Reference: Optional

Feature Under Test: Video Source Configurations

Test Purpose: To verify that Client is able to retrieve a specific video source configuration from Device by using the GetVideoSourceConfiguration operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetVideoSourceConfiguration operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetVideoSourceConfiguration request message with specified ConfigurationToken.
2. Device responds with code HTTP 200 OK and GetVideoSourceConfigurationResponse message.

Test Result:

PASS -

- Client **GetVideoSourceConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoSourceConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetVideoSourceConfiguration>" tag after the "<Body>" tag AND
  - [S2] "<GetVideoSourceConfiguration>" includes tag: "<ConfigurationToken>" with non-empty string value AND
  - [S3] Device response contains "HTTP/* 200 OK" AND

- [S4] Device response contains "<GetVideoSourceConfigurationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: VideoSourceConfigurations_GetVideoSourceConfiguration

10.5 GET VIDEO SOURCE CONFIGURATION OPTIONS

Test Label: Video Source Configuration - Get Video Source Configuration Options

Test Case ID: VIDEOSOURCECONFIGURATIONS-3

Profile S Normative Reference: Conditional

Feature Under Test: Get Video Source Configuration Options

Test Purpose: To verify that Client is able to get video source configuration options provided by Device using the **GetVideoSourceConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoSourceConfigurationOptions** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetVideoSourceConfigurationOptions** request message to retrieve video source configuration options for the Device.
2. Device responds with code HTTP 200 OK and **GetVideoSourceConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetVideoSourceConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoSourceConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetVideoSourceConfigurationOptions** AND

- [S2] If it contains **trt:ConfigurationToken** element THEN it has non-empty string value AND
- [S3] If it contains **trt:ProfileToken** element THEN it has non-empty string value AND

- Device response to the **GetVideoSourceConfigurationOptions** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **trt:GetVideoSourceConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_source_configuration.get_video_source_configuration_options

10.6 SET VIDEO SOURCE CONFIGURATION

Test Label: Configure Video Source Configuration - Set Video Source Configuration

Test Case ID: VIDEOSOURCECONFIGURATIONS-4

Profile S Normative Reference: Conditional

Feature Under Test: Set Video Source Configuration

Test Purpose: To verify that Client is able to change video source configuration provided by Device using the **SetVideoSourceConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetVideoSourceConfiguration** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetVideoSourceConfiguration** request message to change video source configuration on the Device.
2. Device responds with code HTTP 200 OK and **SetVideoSourceConfigurationResponse** message.

Test Result:

PASS -

- Client **SetVideoSourceConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetVideoSourceConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:SetVideoSourceConfiguration** AND
  - [S2] **trt:SetVideoSourceConfiguration/trt:Configuration/@token** element has non-empty string value AND
- Device response to the **SetVideoSourceConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:SetVideoSourceConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_source_configuration.set_video_source_configuration

10.7 GET COMPATIBLE VIDEO SOURCE CONFIGURATIONS

Test Label: Configure Video Source Configuration - Get Compatible Video Source Configurations

Test Case ID: VIDEOSOURCECONFIGURATIONS-5

Profile S Normative Reference: Conditional

Feature Under Test: Get Compatible Video Source Configurations

Test Purpose: To verify that Client is able to retrieve list of video source configurations that are compatible with a certain media profile using the **GetCompatibleVideoSourceConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetCompatibleVideoSourceConfigurations** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetCompatibleVideoSourceConfigurations** request message to retrive compatible video source configurations from the Device.
2. Device responds with code HTTP 200 OK and **GetCompatibleVideoSourceConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetCompatibleVideoSourceConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetCompatibleVideoSourceConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetCompatibleVideoSourceConfigurations** AND
  - [S2] **trt:GetCompatibleVideoSourceConfigurations/trt:ProfileToken** element has non-empty string value AND
- Device response to the **GetCompatibleVideoSourceConfigurations** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:GetCompatibleVideoSourceConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
video_source_configuration.get_compatible_video_source_configurations

10.8 ADD VIDEO SOURCE CONFIGURATION

Test Label: Video Source Configuration - add video source configuration

Test Case ID: VIDEOSOURCECONFIGURATION-6

Profile S Normative Reference: Conditional

Feature Under Test: Video Source Configurations

Test Purpose: To verify that Client is able to add a new or replace an existing video source configuration on Device by using the AddVideoSourceConfiguration operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with AddVideoSourceConfiguration operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes AddVideoSourceConfiguration request message with specified ProfileToken and ConfigurationToken elements to add an existing video source configuration to an existing media profile.
2. Device responds with code HTTP 200 OK and AddVideoSourceConfigurationResponse message.

Test Result:

PASS -

- Client **AddVideoSourceConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddVideoSourceConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<AddVideoSourceConfiguration>" tag after the "<Body>" tag AND
  - [S2] "<AddVideoSourceConfiguration>" includes tag: "<ProfileToken>" with non-empty string value of specific profile token AND
  - [S3] "<AddVideoSourceConfiguration>" includes tag: "<ConfigurationToken>" with non-empty string value of specific configuration token AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<AddVideoSourceConfigurationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: VideoSourceConfigurations_AddVideoSourceConfiguration

11 PTZ - Listing Test Cases

11.1 Expected Scenarios Under Test:

1. Client connects to Device to read PTZ capabilities.
2. Client is considered as supporting PTZ - Listing if the following conditions are met:
   - Client is able to read PTZ capabilities from PTZ Node using EITHER GetNodes OR GetNode operations.
3. Client is considered as NOT supporting PTZ - Listing if ANY of the following is TRUE:
   - No Valid Device Response to GetNodes request AND
   - No Valid Device Response to GetNode request.

11.2 GET NODES

Test Label: PTZ Listing - GetNodes

Test Case ID: PTZLISTING-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Listing

Test Purpose: To verify that list of all existing PTZ capabilities from Device is received by Client using the GetNodes operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetNodes operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetNodes request message to retrieve complete PTZ capabilities list from Device.
2. Device responds with code HTTP 200 OK and GetNodesResponse message.

Test Result:

PASS -

- Client **GetNodes** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetNodes** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetNodes>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetNodesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZListing_GetNodes

11.3 GET NODE

Test Label: PTZ Listing - GetNode

Test Case ID: PTZLISTING-2

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Listing

Test Purpose: To verify that Client is able to retrieve a specific PTZ capability properties from Device using the GetNode operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetNode operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetNode request message to retrieve a specific PTZ capability properties from Device.
2. Device responds with code HTTP 200 OK and GetNodeResponse message.

Test Result:

PASS -

- Client **GetNode** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetNode** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetNode>" tag after the "<Body>" tag AND
  - [S2] "<GetNode>" includes tag: "<NodeToken>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetNodeResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZListing_GetNode

12 PTZ - Configuration Test Cases

12.1 Expected Scenarios Under Test:

1. Client connects to Device to add PTZ configuration to a media profile.
2. Client is considered as supporting PTZ - Configuration if the following conditions are met:
   - Client is able to add PTZ configuration to an existing media profile using GetConfigurations operation AND AddPTZConfiguration operation.
3. Client is considered as NOT supporting PTZ - Configuration if ANY of the following is TRUE:
   - No Valid Device Response to GetConfigurations request OR
   - No Valid Device Response to AddPTZConfiguration request.

12.2 ADD PTZ CONFIGURATION

Test Label: PTZ Configuration - Add PTZ Configuration

Test Case ID: PTZCONFIGURATION-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Configuration

Test Purpose: To verify that Client is able to add PTZ configuration to a profile using GetConfigurations and AddPTZConfiguration operations.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetConfigurations and AddPTZConfiguration operations present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetConfigurations request message to retrieve all available PTZ configurations from Device.
2. Device responds with code HTTP 200 OK and GetConfigurationsResponse message.
3. Client invokes AddPTZConfiguration request message to add a PTZ configuration to an existing media profile.
4. Device responds with code HTTP 200 OK and AddPTZConfigurationResponse message.

Test Result:

PASS -

- Client **GetConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetConfigurations>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetConfigurationsResponse>" tag AND
- Client **AddPTZConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddPTZConfiguration** request in Test Procedure fulfills the following requirements:
  - [S4] Client request contains "<AddPTZConfiguration>" tag after the "<Body>" tag AND
  - [S5] "<AddPTZConfiguration>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S6] "<AddPTZConfiguration>" includes tag: "<ConfigurationToken>" with non-empty string value of specific token AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<AddPTZConfigurationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZConfiguration_AddPTZConfiguration

13 PTZ - Continuous Positioning Test Cases

13.1 Expected Scenarios Under Test:

1. Client connects to Device to control PTZ position using continuous move.
2. Client is considered as supporting PTZ - Continuous Positioning if the following conditions are met:
   - Client is able to move PTZ Device using the ContinuousMove operation with specified PanTilt element OR with specified Zoom element AND
   - Client is able to move PTZ Device using the ContinuousMove operation with specified PanTilt element if Device supports PTZContinuousPanTilt AND
   - Client is able to move PTZ Device using the ContinuousMove operation with specified Zoom element if Device supports PTZContinuousZoom AND
   - Client is able to stop PTZ Device movement using the Stop operation OR using ContinuousMove operation with zero values in PanTilt element if device supports PTZContinuousPanTilt and with zero value in Zoom element if device supports PTZContinuousZoom.
3. Client is considered as NOT supporting PTZ - Continuous Positioning if ANY of the following is TRUE:
   - Client is unable to to move a PTZ device using the ContinuousMove operation with specified EITHER PanTilt element OR Zoom element OR
   - No Valid Device Response to **ContinuousMove** request with specified PanTilt element if device supports PTZContinuousPanTilt OR
   - No Valid Device Response to **ContinuousMove** request with specified Zoom element if device supports PTZContinuousZoom OR
   - Client is unable to stop PTZ movement using EITHER Stop operation OR using ContinuousMove operation OR
   - No Valid Device Response to **Stop** request if detected OR
   - No Valid Device Response to **ContinuousMove** request with zero "x" and "y" attributes values in PanTilt element if detected and if device supports PTZContinuousPanTilt OR
   - No Valid Device Response to **ContinuousMove** request with zero "x" attribute value in Zoom element if detected and if device supports PTZContinuousZoom.

13.2 PTZ CONTINUOUS MOVE PAN/TILT

Test Label: PTZ Continuous Positioning - ContinuousMove PanTilt

Test Case ID: PTZCONTINUOUSPOSITIONING-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Continuous Positioning

Test Purpose: To verify that Client is able to move a PTZ Device using the ContinuousMove operation with specified PanTilt element.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with ContinuousMove operation present.
- Device supports PTZContinuousPanTilt.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes ContinuousMove request message to start move of PTZ Device using specific value of PanTilt element.
2. Device responds with code HTTP 200 OK and ContinuousMoveResponse message.

Test Result:

PASS -

- Client **ContinuousMove** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **ContinuousMove** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<ContinuousMove>" tag after the "<Body>" tag AND
  - [S2] "<ContinuousMove>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S4] "<Velocity>" includes tag: "<PanTilt>" AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<ContinuousMoveResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZContinuousPositioning_ContinuousMovePanTilt

13.3 PTZ CONTINUOUS MOVE ZOOM

Test Label: PTZ Continuous Positioning - ContinuousMove Zoom

Test Case ID: PTZCONTINUOUSPOSITIONING-2

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Continuous Positioning

Test Purpose: To verify that Client is able to change zoom of PTZ Device using the ContinuousMove operation with specified Zoom element.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with ContinuousMove operation present.
- Device supports PTZContinuousZoom.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes ContinuousMove request message to change zoom of PTZ Device using specific value of Zoom element.
2. Device responds with code HTTP 200 OK and ContinuousMoveResponse message.

Test Result:

PASS -

- Client **ContinuousMove** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **ContinuousMove** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<ContinuousMove>" tag after the "<Body>" tag AND
  - [S2] "<ContinuousMove>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S4] "<Velocity>" includes tag: "<Zoom>" AND

- [S6] Device response contains "HTTP/* 200 OK" AND
- [S7] Device response contains "<ContinuousMoveResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZContinuousPositioning_ContinuousMoveZoom

13.4 PTZ STOP

Test Label: PTZ Continuous Positioning - Stop

Test Case ID: PTZCONTINUOUSPOSITIONING-3

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Continuous Positioning

Test Purpose: To verify that Client is able to stop a PTZ Device movement using the Stop operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with Stop operation present

Test Procedure (expected to be reflected in network trace file):

1. Client invokes Stop request message to stop ongoing movements of PTZ Device.
2. Device responds with code HTTP 200 OK and StopResponse message.

Test Result:

PASS -

- Client **Stop** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Stop** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tptz:Stop** AND
  - [S2] **tptz:Stop/tptz:ProfileToken** element has non-empty string value of specific token AND

- Device response on the **Stop** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tptz:StopResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZContinuousPositioning_Stop

13.5 STOP MOVEMENT USING PTZ CONTINUOUS MOVE

Test Label: PTZ Continuous Positioning - Stop Movement using ContinuousMove

Test Case ID: PTZCONTINUOUSPOSITIONING-4

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Continuous Positioning

Test Purpose: To verify that Client is able to stop a PTZ Device movement using ContinuousMove operation with zero values in PanTilt and Zoom elements.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with ContinuousMove operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes ContinuousMove request message with zero "x" and "y" attributes values in PanTilt element.
2. Client invokes ContinuousMove request message with zero "x" attribute value in Zoom element.
3. Device responds with code HTTP 200 OK and ContinuousMoveResponse message.

Test Result:

NOTE: In case when device does not support both PTZContinuousPanTilt and PTZContinuousZoom features then the test shall be deemed as "NOT DETECTED". In case Client does not send ContinuousMove request message with zero "x" and "y" attributes values in PanTilt element if device supports PTZContinuousPanTilt then the test shall be deemed as "NOT

DETECTED". In case Client does not send ContinuousMove request message with zero "x" attribute value in Zoom element if device supports PTZContinuousZoom then the test shall be deemed as "NOT DETECTED".

PASS -

- If device supports PTZContinuousPanTilt then there is client **ContinuousMove** request messages which corresponds to the following requirements (else skip the check):
  - Client **ContinuousMove** request messages are valid according to XML Schemas listed in Namespaces AND
  - Client **ContinuousMove** request in Test Procedure fulfills the following requirements:
    - [S1] **soapenv:Body** element has child element **tptz:ContinuousMove** AND
    - [S2] **tptz:ContinuousMove/tptz:ProfileToken** element has non-empty string value of specific token AND
    - [S3] **tptz:ContinuousMove/tptz:Velocity** containt tag **tt:PanTilt** AND
    - [S4] **tptz:ContinuousMove/tptz:Velocity/tt:PanTilt/@x** attribute value is equal to 0 AND
    - [S5] **tptz:ContinuousMove/tptz:Velocity/tt:PanTilt/@y** attribute value is equal to 0 AND
  - Device response on the **ContinuousMove** request fulfills the following requirements:
    - [S6] It has HTTP 200 response code AND
    - [S7] **soapenv:Body** element has child element **tptz:ContinuousMoveResponse**.
- If device supports PTZContinuousZoom then there is client **ContinuousMove** request messages which corresponds to the following requirements (else skip the check):
  - Client **ContinuousMove** request messages are valid according to XML Schemas listed in Namespaces AND
  - Client **ContinuousMove** request in Test Procedure fulfills the following requirements:
    - [S8] **soapenv:Body** element has child element **tptz:ContinuousMove** AND
    - [S9] **tptz:ContinuousMove/tptz:ProfileToken** element has non-empty string value of specific token AND
    - [S10] **tptz:ContinuousMove/tptz:Velocity** containt tag **tt:Zoom** AND

- [S11] **tptz:ContinuousMove/tptz:Velocity/tt:Zoom/@x** attribute value is equal to 0.

- Device response on the **ContinuousMove** request fulfills the following requirements:
  - [S12] It has HTTP 200 response code AND
  - [S13] **soapenv:Body** element has child element **tptz:ContinuousMoveResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZContinuousPositioning_StopMovementUsingContinuousMove

14 PTZ - Absolute Positioning Test Cases

14.1 Expected Scenarios Under Test:

1. Client connects to Device to read PTZ capabilities and control the position using absolute positioning.
2. Client is considered as supporting PTZ - Absolute Positioning if the following conditions are met:
   - Client is able to read PTZ capabilities from PTZ Node using EITHER GetNodes OR GetNode operations AND
   - Client is able to move PTZ Device using the AbsoluteMove operation by EITHER Move a PTZ Device using the AbsoluteMove operation with specified PanTilt element OR change zoom of PTZ Device using the AbsoluteMove operation with specified Zoom element.
3. Client is considered as NOT supporting PTZ - Absolute Positioning if ANY of the following is TRUE:
   - BOTH (No Valid Device Response to GetNodes request AND No Valid Device Response to GetNode request) OR
   - BOTH (No Valid Device Response to AbsoluteMove request with specified PanTilt element AND No Valid Device Response to AbsoluteMove request with specified Zoom element).

14.2 PTZ ABSOLUTE MOVE PAN/TILT

Test Label: PTZ Absolute Positioning - AbsoluteMove PanTilt

Test Case ID: PTZABSOLUTEPOSITIONING-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Absolute Positioning

Test Purpose: To verify that Client is able to move a PTZ Device using the AbsoluteMove operation with specified PanTilt element.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with AbsoluteMove operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes AbsoluteMove request message to move of PTZ Device using specific value of PanTilt element.
2. Device responds with code HTTP 200 OK and AbsoluteMoveResponse message.

Test Result:

NOTE: If Client AbsoluteMove request message does not contain "<PanTilt>" tag inside "<Position>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

- Client **AbsoluteMove** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AbsoluteMove** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<AbsoluteMove>" tag after the "<Body>" tag AND
  - [S2] "<AbsoluteMove>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S4] "<Position>" includes tag: "<PanTilt>" AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<AbsoluteMoveResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZAbsolutePositioning_AbsoluteMovePanTilt

14.3 PTZ ABSOLUTE MOVE ZOOM

Test Label: PTZ Absolute Positioning - AbsoluteMove Zoom

Test Case ID: PTZABSOLUTEPOSITIONING-2

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Absolute Positioning

Test Purpose: To verify that Client is able to change zoom of PTZ Device using the AbsoluteMove operation with specified Zoom element.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with AbsoluteMove operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes AbsoluteMove request message to change zoom of PTZ Device using specific value of Zoom element.
2. Device responds with code HTTP 200 OK and AbsoluteMoveResponse message.

Test Result:

PASS -

- Client **AbsoluteMove** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AbsoluteMove** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<AbsoluteMove>" tag after the "<Body>" tag AND
  - [S2] "<AbsoluteMove>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S4] "<Position>" includes tag: "<Zoom>" AND
  - [S6] Device response contains "HTTP/* 200 OK" AND
  - [S7] Device response contains "<AbsoluteMoveResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZAbsolutePositioning_AbsoluteMoveZoom

15 PTZ - Relative Positioning Test Cases

15.1 Expected Scenarios Under Test:

1. Client connects to Device to read PTZ capabilities and control the position using relative positioning.
2. Client is considered as supporting PTZ - Relative Positioning if the following conditions are met:
   - Client is able to read PTZ capabilities from PTZ Node using EITHER GetNodes OR GetNode operations AND
   - Client is able to move PTZ Device using the RelativeMove operation by EITHER Move a PTZ Device using the RelativeMove operation with specified PanTilt element OR change zoom of PTZ Device using the RelativeMove operation with specified Zoom element).
3. Client is considered as NOT supporting PTZ - Relative Positioning if ANY of the following is TRUE:
   - BOTH (No Valid Device Response to GetNodes request AND No Valid Device Response to GetNode request) OR
   - BOTH (No Valid Device Response to RelativeMove request with specified PanTilt element AND No Valid Device Response to RelativeMove request with specified Zoom element).

15.2 PTZ RELATIVE MOVE PAN/TILT

Test Label: PTZ Relative Positioning - RelativeMove PanTilt

Test Case ID: PTZRELATIVEPOSITIONING-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Relative Positioning

Test Purpose: To verify that Client is able to move a PTZ Device using the RelativeMove operation with specified PanTilt element.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with RelativeMove operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes RelativeMove request message to move of PTZ Device using specific value of PanTilt element.
2. Device responds with code HTTP 200 OK and RelativeMoveResponse message.

Test Result:

NOTE: If Client RelativeMove request message does not contain "<PanTilt>" tag inside "<Translation>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

- Client **RelativeMove** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **RelativeMove** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<RelativeMove>" tag after the "<Body>" tag AND
  - [S2] "<RelativeMove>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S4] "<Translation>" includes tag: "<PanTilt>" AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<RelativeMoveResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZRelativePositioning_RelativeMovePanTilt

15.3 PTZ RELATIVE MOVE ZOOM

Test Label: PTZ Relative Positioning - RelativeMove Zoom

Test Case ID: PTZRELATIVEPOSITIONING-2

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Relative Positioning

Test Purpose: To verify that Client is able to change zoom of PTZ Device using the RelativeMove operation with specified Zoom element.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with RelativeMove operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes RelativeMove request message to change zoom of PTZ Device using specific value of Zoom element.
2. Device responds with code HTTP 200 OK and RelativeMoveResponse message.

Test Result:

NOTE: If Client AbsoluteMove request message does not contain "<Zoom>" tag inside "<Translation>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

- Client **RelativeMove** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **RelativeMove** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<RelativeMove>" tag after the "<Body>" tag AND
  - [S2] "<RelativeMove>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S4] "<Translation>" includes tag: "<Zoom>" AND
  - [S6] Device response contains "HTTP/* 200 OK" AND
  - [S7] Device response contains "<RelativeMoveResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZRelativePositioning_RelativeMoveZoom

16 PTZ - Presets Test Cases

16.1 Expected Scenarios Under Test:

1. Client connects to Device to manage the presets of a PTZ Node.
2. Client is considered as supporting PTZ - Presets if the following conditions are met:
   - Client is able to list the presets using the GetPresets operation AND
   - Client is able to move a PTZ Device to a specific preset using the GotoPreset operation.
3. Client is considered as NOT supporting PTZ - Presets if ANY of the following is TRUE:
   - No Valid Device Response to GetPresets request OR
   - No Valid Device Response to GotoPreset request.

16.2 PTZ GET PRESETS

Test Label: PTZ Presets - GetPresets

Test Case ID: PTZPRESETS-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Presets

Test Purpose: To verify that Client is able to list the presets using the GetPresets operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetPresets operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetPresets request message to list the available presets from Device.
2. Device responds with code HTTP 200 OK and GetPresetsResponse message.

Test Result:

PASS -

- Client **GetPresets** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetPresets** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetPresets>" tag after the "<Body>" tag AND
  - [S2] "<GetPresets>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetPresetsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZPresets_GetPresets

16.3 PTZ GOTO PRESET

Test Label: PTZ Presets - GotoPreset

Test Case ID: PTZPRESETS-2

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Presets

Test Purpose: To verify that Client is able to move a PTZ Device to a specific preset using the GotoPreset operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GotoPreset operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GotoPreset request message to move PTZ Device to specific preset.
2. Device responds with code HTTP 200 OK and GotoPresetResponse message.

Test Result:

PASS -

- Client **GotoPreset** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GotoPreset** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GotoPreset>" tag after the "<Body>" tag AND
  - [S2] "<GotoPreset>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S3] "<GotoPreset>" includes tag: "<PresetToken>" with non-empty string value of specific token AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<GotoPresetResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZPresets_GotoPreset

17 PTZ - Home Position Test Cases

17.1 Expected Scenarios Under Test:

1. Client connects to Device to manage the home position of a PTZ Node.
2. Client is considered as supporting PTZ - Home Position if the following conditions are met:
   - Client is able to move PTZ Device to its home position using the GotoHomePosition operation
3. Client is considered as NOT supporting PTZ - Home Position if ANY of the following is TRUE:
   - No Valid Device Response to GotoHomePosition request.

17.2 PTZ HOME POSITION

Test Label: PTZ Presets - GotoHomePosition

Test Case ID: PTZHOMEPOSITION-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Presets

Test Purpose: To verify that Client is able to move PTZ Device to its home position using the GotoHomePosition operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GotoHomePosition operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GotoHomePosition request message to move PTZ Device to its home position.
2. Device responds with code HTTP 200 OK and GotoHomeResponse message.

Test Result:

PASS -

- Client **GotoHomePosition** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GotoHomePosition** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GotoHomePosition>" tag after the "<Body>" tag AND
  - [S2] "<GotoHomePosition>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GotoHomePositionResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: PTZPresets_GotoHomePosition

18 PTZ - Auxiliary Command Test Cases

18.1 Feature Level Requirement:

Validated Feature: ptz.auxiliary

Profile S Requirement: Conditional

18.2 Expected Scenarios Under Test:

1. Client connects to Device to manage the auxiliary commands of a PTZ Node.
2. Client is considered as supporting PTZ - Auxiliary Command if the following conditions are met:
   - Client is able to read PTZ capabilities from PTZ Node using EITHER GetNodes OR GetNode operations AND
   - Client is able to call an auxiliary operation on Device using the SendAuxiliaryCommand operation.
3. Client is considered as NOT supporting PTZ - Auxiliary Command if ANY of the following is TRUE:
   - BOTH (No Valid Device Response to GetNodes request AND No Valid Device Response to GetNode request) OR
   - No Valid Device Response to SendAuxiliaryCommand request.

18.3 PTZ SEND AUXILIARY COMMAND

Test Label: PTZ Auxiliary Command - Send Auxiliary Command

Test Case ID: PTZAUXILIARYCOMMAND-1

Profile S Normative Reference: Conditional

Feature Under Test: PTZ Auxiliary Command

Test Purpose: To verify that Client is able to call an auxiliary operation on Device using the **SendAuxiliaryCommand** operation (PTZ Service).

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with **SendAuxiliaryCommand** operation (PTZ Service) present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SendAuxiliaryCommand** request message (PTZ Service) to call an auxiliary operation on Device.
2. Device responds with code HTTP 200 OK and **SendAuxiliaryCommandResponse** message.

Test Result:

PASS -

- Client **SendAuxiliaryCommand** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SendAuxiliaryCommand** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tptz:SendAuxiliaryCommand** AND
  - [S2] It contains **tptz:ProfileToken** element with non-empty string value AND
  - [S3] It contains **tptz:AuxiliaryData** element with non-empty string value AND
- Device response on the **SendAuxiliaryCommand** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **tptz:SendAuxiliaryCommandResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ptz.auxiliary.send_auxiliary_command

19 Audio Streaming Test Cases

19.1 Feature Level Requirement:

Validated Feature: AudioStreaming

Profile S Requirement: Conditional

19.2 Expected Scenarios Under Test:

1. Client connects to Device to configure a media profile and initiate Audio Streaming with G.711 encoding type.
2. Client is considered as supporting Audio Streaming if the following conditions are met:
   - Client is able to configure a media profile for audio streaming using the GetCompatibleAudioSourceConfigurations, AddAudioSourceConfiguration, GetCompatibleAudioEncoderConfigurations and AddAudioEncoderConfiguration operations AND
   - Client is able to initiate and retrieve audio stream with G.711 encoding type AND
   - When Device and Client support G.726 encoding type for Audio Streaming:
     - Client is able to initiate and retrieve audio stream with G.726 encoding type AND
   - When Device and Client support AAC encoding type for Audio Streaming:
     - Client is able to initiate and retrieve audio stream with AAC encoding type.
3. Client is considered as NOT supporting Audio Streaming if ANY of the following is TRUE:
   - No Valid Device Response to GetCompatibleAudioSourceConfigurations request OR
   - No Valid Device Response to AddAudioSourceConfiguration request OR
   - No Valid Device Response to GetCompatibleAudioEncoderConfigurations request OR
   - No Valid Device Response to AddAudioEncoderConfiguration request OR
   - G.711 Audio Streaming attempts detected have failed OR
   - When Device and Client support G.726 encoding type for Audio Streaming:
     - Client is unable to initiate and retrieve audio stream with G.726 encoding type OR

- When Device and Client support AAC encoding type for Audio Streaming:
  - Client is unable to initiate and retrieve audio stream with AAC encoding type.

19.3 CONFIGURE MEDIA PROFILE FOR AUDIO STREAMING

Test Label: Audio Streaming - Configure Media Profile

Test Case ID: AUDIOSTREAMING-1

Profile S Normative Reference: Conditional

Feature Under Test: Audio Streaming

Test Purpose: To verify that Client is able to to configure a media profile for audio streaming using the GetCompatibleAudioSourceConfigurations, AddAudioSourceConfiguration, GetCompatibleAudioEncoderConfigurations and AddAudioEncoderConfiguration operations.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetCompatibleAudioSourceConfigurations, AddAudioSourceConfiguration, GetCompatibleAudioEncoderConfigurations and AddAudioEncoderConfiguration operations present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetCompatibleAudioSourceConfigurations request message to retrieve all audio source configurations of Device that are compatible with a certain media profile.
2. Device responds with code HTTP 200 OK and GetCompatibleAudioSourceConfigurationsResponse message.
3. Client invokes AddAudioSourceConfiguration request message to add audio source configuration to an existing media profile.
4. Device responds with code HTTP 200 OK and AddAudioSourceConfigurationResponse message.
5. Client invokes GetCompatibleAudioEncoderConfigurations request message to retrieve all audio encoder configurations of the device that are compatible with a certain media profile.
6. Device responds with code HTTP 200 OK and GetCompatibleAudioEncoderConfigurationsResponse message.

7. Client invokes AddAudioEncoderConfiguration request message to add audio encoder configuration to an existing media profile.

8. Device responds with code HTTP 200 OK and AddAudioEncoderConfigurationResponse message.

Test Result:

PASS -

- Client GetCompatibleAudioSourceConfigurations request message is a well-formed SOAP request (refer to onvif.xsd) AND
- Client GetCompatibleAudioSourceConfigurations request message has a proper hierarchy (refer to media.wsdl) AND
  - [S1] Client request contains "<GetCompatibleAudioSourceConfigurations>" tag after the "<Body>" tag AND
  - [S2] "<GetCompatibleAudioSourceConfigurations>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetCompatibleAudioSourceConfigurationsResponse>" tag AND
- Client AddAudioSourceConfiguration request message is a well-formed SOAP request (refer to onvif.xsd) AND
- Client AddAudioSourceConfiguration request message has a proper hierarchy (refer to media.wsdl) AND
  - [S5] Client request contains "<AddAudioSourceConfiguration>" tag after the "<Body>" tag AND
  - [S6] "<AddAudioSourceConfiguration>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S7] "<AddAudioSourceConfiguration>" includes tag: "<ConfigurationToken>" with non-empty string value of specific token AND
  - [S8] Device response contains "HTTP/* 200 OK" AND
  - [S9] Device response contains "<AddAudioSourceConfigurationResponse>" tag AND

- Client GetCompatibleAudioEncoderConfigurations request message is a well-formed SOAP request (refer to onvif.xsd) AND
- Client GetCompatibleAudioEncoderConfigurations request message has a proper hierarchy (refer to media.wsdl) AND
  - [S10] Client request contains "<GetCompatibleAudioEncoderConfigurations>" tag after the "<Body>" tag AND
  - [S12] "<GetCompatibleAudioEncoderConfigurations>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S13] Device response contains "HTTP/* 200 OK" AND
  - [S14] Device response contains "<GetCompatibleAudioEncoderConfigurationsResponse>" tag AND
- Client AddAudioEncoderConfiguration request message is a well-formed SOAP request (refer to onvif.xsd) AND
- Client AddAudioEncoderConfiguration request message has a proper hierarchy (refer to media.wsdl) AND
  - [S15] Client request contains "<AddAudioEncoderConfiguration>" tag after the "<Body>" tag AND
  - [S16] "<AddAudioEncoderConfiguration>" includes tag: "<ProfileToken>" with non-empty string value of specific token AND
  - [S17] "<AddAudioEncoderConfiguration>" includes tag: "<ConfigurationToken>" with non-empty string value of specific token AND
  - [S18] Device response contains "HTTP/* 200 OK" AND
  - [S19] Device response contains "<AddAudioEncoderConfigurationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: audio_streaming.configure_media_profile

19.4 G.711 AUDIO STREAMING

Test Label: Audio Streaming - G.711

Test Case ID: AUDIOSTREAMING-2

Profile S Normative Reference: Conditional

Feature Under Test: Audio Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve audio stream with G.711 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with Audio Streaming of G.711 encoding type.
- Device supports G.711 encoding for Audio streaming.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Source Configuration and Audio Encoder Configuration with G711 Encoding value. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "audio" and with encoding name "PCMU" or with payload type number "0".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to set media session parameters for G711 audio streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] IF SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND with sessions attribute "rtpmap" THEN encoding name is "PCMU"
  - [S3] ELSE IF SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND without sessions attribute "rtpmap" THEN payload type number is "0" (see [RFC 3551]) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S5] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S6] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S7] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S8] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S9] It has HTTP 200 response code AND
  - [S10] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It received before the Client **RTSP DESCRIBE** request AND
  - [S12] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S13] It invoked for the same Device as for the Client **RTSP SETUP** request AND

- [S14] It invoked after the Client **RTSP SETUP** request AND
- [S15] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S17] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S18] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S19] It invoked after the Client **RTSP PLAY** request AND
  - [S20] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: audio_streaming.g711

19.5 G.726 AUDIO STREAMING

Test Label: Audio Streaming - G.726

Test Case ID: AUDIOSTREAMING-3

Profile S Normative Reference: Conditional

Feature Under Test: Audio Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve audio stream with G.726 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Audio Streaming of G.726 encoding type.
- Device supports G.726 encoding for Audio streaming.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Source Configuration and Audio Encoder Configuration with G726 Encoding value. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "audio" and with encoding name "G726-*".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for G.726 audio streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND

- [S2] SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND with sessions attribute "rtpmap" with encoding name "G726-*" (see [RFC 3551]) AND

- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:

- [S17] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S18] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S19] It invoked after the Client **RTSP PLAY** request AND
  - [S20] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: audio_streaming.g726

19.6 AAC AUDIO STREAMING

Test Label: Audio Streaming - AAC

Test Case ID: AUDIOSTREAMING-4

Profile S Normative Reference: Conditional

Feature Under Test: Audio Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve audio stream with AAC encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Audio Streaming of AAC encoding type.
- Device supports AAC encoding for Audio streaming.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Source Configuration and Audio Encoder Configuration with AAC Encoding value. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "audio" and with encoding name "MPEG4-GENERIC".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for AAC audio streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND with sessions attribute "rtpmap" with encoding name "MPEG4-GENERIC" (see [RFC 3640]) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND

- [S4] It invoked after the Client **RTSP DESCRIBE** request AND
- [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND

- [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:

  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: audio_streaming.aac

20 Metadata Configurations Test Cases

20.1 Feature Level Requirement:

Validated Feature: MetadataConfigurations

Profile S Requirement: Conditional

20.2 Expected Scenarios Under Test:

1. Client connects to Device to list and modify Metadata Configurations.
2. Client is considered as supporting Metadata Configurations if the following conditions are met:
   - Client is able to get metadata parameter options by using **GetMetadataConfigurationOptions** operation AND
   - Client is able to modify an existing metadata configuration by using **SetMetadataConfiguration** command.
3. Client is considered as NOT supporting Metadata Configurations if ANY of the following is TRUE:
   - No Valid Device Response to **GetMetadataConfigurations** request if detected OR
   - No Valid Device Response to **GetMetadataConfiguration** request if detected OR
   - No Valid Device Response to **GetMetadataConfigurationOptions** request OR
   - No Valid Device Response to **SetMetadataConfiguration** request.

20.3 LIST METADATA CONFIGURATIONS

Test Label: Metadata Configurations - List All Existing Metadata Configurations

Test Case ID: METADATACONFIGURATIONS-1

Profile S Normative Reference: Optional

Feature Under Test: Metadata Configurations

Test Purpose: To verify that list of all existing metadata configurations from Device is received by Client using the **GetMetadataConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfigurations** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfigurations** request message to retrieve complete list of available metadata configurations from Device.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetMetadataConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMetadataConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetMetadataConfigurations** AND
- Device response on the **GetMetadataConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **trt:GetMetadataConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MetadataConfigurations_GetMetadataConfigurations

20.4 GET SPECIFIC METADATA CONFIGURATION

Test Label: Metadata Configurations - Gets a Specific Metadata Configuration

Test Case ID: METADATACONFIGURATIONS-2

Profile S Normative Reference: Optional

Feature Under Test: Metadata Configurations

Test Purpose: To verify that Client is able to retrieve a specific metadata configuration from Device by using the **GetMetadataConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfiguration** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfiguration** request message with specified ConfigurationToken.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationResponse**.

Test Result:

PASS -

- Client **GetMetadataConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMetadataConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetMetadataConfiguration** AND
  - [S2] **trc:GetMetadataConfiguration/trt:ConfigurationToken** element has non-empty string value of specific metadata configuraton token AND
- Device response on the **GetMetadataConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:GetMetadataConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MetadataConfigurations_GetMetadataConfiguration

20.5 GET METADATA CONFIGURATION OPTIONS

Test Label: Metadata Configurations - Get Metadata Configuration Options

Test Case ID: METADATACONFIGURATIONS-3

Profile S Normative Reference: Conditional

Feature Under Test: Metadata Configurations

Test Purpose: To verify that Client is able to get metadata configuration options by using **GetMetadataConfigurationOptions** operation

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfigurationOptions** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfigurationOptions** request message to retrieve supported metadata configuration options from Device.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetMetadataConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMetadataConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetMetadataConfigurationOptions** AND
  - If it contains **trt:GetMetadataConfigurationOptions/trt:ConfigurationToken** element then it fulfills the following requirements (else skip the check):
    - [S2] **trc:GetMetadataConfigurationOptions/trt:ConfigurationToken** element has non-empty string value of specific metadata configuraton token AND
  - If it contains **trt:GetMetadataConfigurationOptions/trt:ProfileToken** element then it fulfills the following requirements (else skip the check):
    - [S3] **trc:GetMetadataConfigurationOptions/trt:ProfileToken** element has non-empty string value of specific profile token AND

- Device response on the **GetMetadataConfigurationOptions** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **trt:GetMetadataConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MetadataConfigurations_GetMetadataConfigurationOptions

20.6 MODIFY METADATA CONFIGURATION

Test Label: Metadata Configurations - Modify Metadata Configuration

Test Case ID: METADATACONFIGURATIONS-4

Profile S Normative Reference: Conditional

Feature Under Test: Metadata Configurations

Test Purpose: To verify that Client is able to modify metadata configuration on Device by using the **SetMetadataConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetMetadataConfiguration** operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetMetadataConfiguration** request message to change metadata configuration settings with any modifications of parameters.
2. Device responds with code HTTP 200 OK and **SetMetadataConfigurationResponse** message.

Test Result:

PASS -

- Client **SetMetadataConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **SetMetadataConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:SetMetadataConfiguration** AND
  - [S2] **trt:SetMetadataConfiguration/trt:Configuration/@token** attribute has non-empty string value of specific configuration token AND
- Device response on the **SetMetadataConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:SetMetadataConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MetadataConfigurations_ModifyMetadataConfiguration

21 Multiple Video Sources Test Cases

21.1 Feature Level Requirement:

Validated Feature: Multiple Video Sources

Profile S Requirement: Mandatory

21.2 Expected Scenarios Under Test:

1. Client connects to Device to get all Video Sources.
2. Client obtains video streaming for each Video Source provided by a Device.
3. Client is considered as supporting Multiple Video Sources if the following conditions are met:
   - Client is able to get profile list by using **GetProfiles** operation AND
   - Client is able to to initiate and retrieve video stream for each Video Source provided by a Device using **GetStreamUri** command and **RTSP** commands.
4. Client is considered as NOT supporting Multiple Video Sources if ANY of the following is TRUE:
   - No Valid Device Response to **GetProfiles** request OR
   - Client is unable to initiate and retrieve video streaming for at least one Video Source provided by a Device.

21.3 STREAMING WITH ALL VIDEO SOURCES DETECTED IN GET PROFILES

Test Label: Multiple Video Sources - Streaming with all Video Sources detected in GetProfilesResponse

Test Case ID: MULTIPLEVIDEOSOURCES-1

Profile S Normative Reference: Mandatory

Feature Under Test: Multiple Video Sources

Test Purpose: To verify that Client is able to obtaine video streaming for each video source provided by a Device in GetProfiles responses.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with video streaming present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetProfiles** request messages to retrieve complete list of available media profiles with video source configurations from Device.
2. Device responds with code HTTP 200 OK and **GetProfilesResponse** message.
3. Client initiate video streaming for each Video Source token detected in GetProfilesResponse:
   - Client selects existing media profile with required Video Source token or modifies media profile to have required Video Source token or creates media profile with required Video Source token.
   - Client invokes **GetStreamUri** request for this media profile.
   - Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
   - Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
   - Device responds with code RTSP 200 OK and SDP information with Media Type: "video".
   - Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for video streaming.
   - Device responds with code RTSP 200 OK.
   - Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
   - Device responds with code RTSP 200 OK.
   - Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
   - If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

PASS -

- For each Video Source Token listed by HelperGetVideoSourcesListFromGetProfiles (see Annex A.1) there is a video stream in Test Procedure that fulfills the following requirements:

- There is a Client **GetStreamUri** request that fulfills the following requirements:
  - [S1] It invoked for the media profile which contains Video Source Configuration with this Video Source Token (see Annex A.2 HelperGetVideoSourceTokenUsedForStreaming to get video source token from media profile) AND
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **trt:GetStreamUriResponse** AND
- There is a RTSP session in Test Procedure that fulfills the following requirements:
  - [S5] It invoked for the uri from **GetStreamUri** response AND
  - [S6] It started video streaming according to HelperFindVideoStreamingForGetStreamUri (see Annex A.3)

FAIL -

- The Client failed PASS criteria.

Validated Feature List: multiple_video_sources.streaming_for_video_sources_from_get_profiles

22 Multiple Audio Sources Test Cases

22.1 Feature Level Requirement:

Validated Feature: Multiple Audio Sources

Profile S Requirement: Conditional

22.2 Expected Scenarios Under Test:

1. Client connects to Device to get all Audio Sources.
2. Client obtains audio streaming for each Audio Source provided by a Device.
3. Client is considered as supporting Multiple Audio Sources if the following conditions are met:
   - Client is able to get profile list by using **GetProfiles** operation AND
   - Client is able to to initiate and retrieve audio stream for each Audio Source provided by a Device using **GetStreamUri** command and **RTSP** commands.
4. Client is considered as NOT supporting Multiple Audio Sources if ANY of the following is TRUE:
   - No Valid Device Response to **GetProfiles** request OR
   - Client is unable to initiate and retrieve audio streaming for at least one Audio Source provided by a Device.

22.3 STREAMING WITH ALL AUDIO SOURCES DETECTED IN GET PROFILES

Test Label: Multiple Audio Sources - Streaming with all Audio Sources detected in GetProfilesResponse

Test Case ID: MULTIPLEAUDIOSOURCES-1

Profile S Normative Reference: Conditional

Feature Under Test: Multiple Audio Sources

Test Purpose: To verify that Client is able to obtaine audio streaming for each audio source provided by a Device in GetProfiles responses.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with audio streaming present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetProfiles** request messages to retrieve complete list of available media profiles with audio source configurations from Device.
2. Device responds with code HTTP 200 OK and **GetProfilesResponse** message.
3. Client initiate audio streaming for each Audio Source token detected in GetProfilesResponse:
   - Client selects existing media profile with required Audio Source token or modifies media profile to have required Audio Source token or creates media profile with required Audio Source token.
   - Client invokes **GetStreamUri** request for this media profile.
   - Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
   - Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
   - Device responds with code RTSP 200 OK and SDP information with Media Type: "audio".
   - Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for audio streaming.
   - Device responds with code RTSP 200 OK.
   - Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
   - Device responds with code RTSP 200 OK.
   - Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
   - If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK.

Test Result:

Note: If Client does not initiate audio streaming then Test shall be deemed as "NOT DETECTED".

PASS -

- For each Audio Source Token listed by HelperGetAudioSourcesListFromGetProfiles (see Annex A.4) there is a audio stream in Test Procedure that fulfills the following requirements:

- There is a Client **GetStreamUri** request that fulfills the following requirements:
  - [S1] It invoked for the media profile which contains Audio Source Configuration with this Audio Source Token (see Annex A.5 HelperGetAudioSourceTokenUsedForStreaming to get audio source token from media profile) AND
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **trt:GetStreamUriResponse** AND
- There is a RTSP session in Test Procedure that fulfills the following requirements:
  - [S5] It invoked for the uri from **GetStreamUri** response AND
  - [S6] It started audio streaming according to HelperFindAudioStreamingForGetStreamUri (see Annex A.6)

FAIL -

- The Client failed PASS criteria.

Validated Feature List: multiple_audio_sources.streaming_for_audio_sources_from_get_profiles

Annex A Test for Appendix A

A.1 Get Video Sources List from GetProfiles responses

Name: HelperGetVideoSourcesListFromGetProfiles

Procedure Purpose: Collect list of video source tokens provided by the device in GetProfiles responses.

Pre-requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetProfiles** operation present.

Input: None

Returns: The complete list of video source tokens detected in all GetProfiles responses (*videoSourcesList*).

Annex Procedure:

- For each **GetProfiles** response detected in the Conversations the Client Test Tool does the following:
  - For each **trt:GetProfilesResponse/trt:Profiles** detected in the Conversations the Client Test Tool does the following:
    - If it contains **VideoSourceConfiguration** element THEN the Client Test Tool adds **tt:VideoSourceConfiguration/tt:SourceToken** value to the *videoSourcesList* if this value does not exists in it.

A.2 Get Video Source Token That was Used for Streaming

Name: HelperGetVideoSourceTokenUsedForStreaming

Procedure Purpose: Get video source token that was used in the media profile requested by the Client in **GetStreamUri** request.

Pre-requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetProfiles** operation or **AddVideoSourceConfiguration** present.

Input: GetStreamUri request

Returns: Video Source token (*videoSourceToken*).

Annex Procedure:

- The Client Test Tool checks that there is Client **AddVideoSourceConfiguration** request or **GetProfiles** response in Test Procedure that fulfills the following requirements:
  - [S1] It is invoked for the same Device as **GetStreamUri** request AND
  - [S2] It is the closest one preceding **GetStreamUri** request and it fullfils [S3] or [S4] requirement AND
  - If it is **AddVideoSourceConfiguration** request:
    - [S3] **trt:AddVideoSourceConfiguration/trt:ProfileToken** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
  - If it is **GetProfiles** response:
    - [S4] It contains **trt:Profiles** element with **trt:Profiles/@token** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
- The Client Test Tool checks if there is **SetVideoSourceConfiguration** command that fulfills the following requirements:
  - [S5] It invoked for the same Device as **GetStreamUri** request AND
  - If **AddVideoSourceConfiguration** request was found during previous steps:
    - [S6] It invoked after **AddVideoSourceConfiguration** request AND
    - [S7] It is the closest one preceding the **GetStreamUri** request AND
    - [S8] **trt:SetVideoSourceConfiguration/trt:Configuration/@token** value is equal to **trt:AddVideoSourceConfiguration/trt:ConfigurationToken** value AND
  - If **GetProfiles** request was found during previous steps:
    - [S9] It invoked after **GetProfiles** request AND
    - [S10] It is the closest one preceding the **GetStreamUri** request AND
    - [S11] **trt:SetVideoSourceConfiguration/trt:Configuration/@token** value is equal to **tt:VideoSourceConfiguration/@token** value from **trt:GetProfilesResponse/ trt:Profiles** with trt:Profiles/@token attribute value is equal to **trt:GetStreamUri/ trt:ProfileToken** value AND

- IF **SetVideoSourceConfiguration** command was detected during previous steps than **trt:SetVideoSourceConfiguration/trt:Configuration/tt:SourceToken** value will be returened as result of current procedure
- [S12] ELSE IF **GetProfiles** request was found during previous steps then **tt:VideoSourceConfiguration/tt:SourceToken** value from **trt:GetProfilesResponse/ trt:Profiles** element with trt:Profiles/@token is equal to **trt:GetStreamUri/trt:ProfileToken** value will be returened as result of current procedure
- ELSE IF **AddVideoSourceConfiguration** request was found during previous steps and no **SetVideoSourceConfiguration** was found during previous steps, the Client Test Tool checks the following:
  - There is **GetCompatibleVideoSourceConfigurations** request in Test Procedure that fulfills the following requirements:
    - [S13] It is invoked for the same Device as the **AddVideoSourceConfiguration** request AND
    - [S14] It is the closest one preceding the **AddVideoSourceConfiguration** request AND
    - [S15] **trt:GetCompatibleVideoSourceConfigurations/trt:ProfileToken** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
  - Device response on the **GetCompatibleVideoSourceConfigurations** request fulfills the following requirements:
    - [S16] It has HTTP 200 response code AND
    - [S17] **soapenv:Body** element has child element **trt:GetCompatibleVideoSourceConfigurationsResponse** AND
    - [S18] It contains **trt:Configurations/@token** value is equal to **trt:AddVideoSourceConfiguration/trt:ConfigurationToken** value AND
  - [S19] **trt:Configurations/tt:SourceToken** value from **trt:GetCompatibleVideoSourceConfigurationsResponse/trt:Configurations** element with @token is equal to **trt:AddVideoSourceConfiguration/trt:ConfigurationToken** value will be returened as result of current procedure

A.3 Find Video Streaming corresponding to GetStreamUri

Name: HelperFindVideoStreamingForGetStreamUri

Procedure Purpose: Find video streaming whcih corresponding to GetStreamUri pair.

Pre-requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with video streaming present.

Input: GetStreamUri

Returns: None.

Annex Procedure:

- There is Client **RTSP DESCRIBE** request in Test Procedure that fulfills the following requirements:
  - [S1] It invoked for the same Device as **GetStreamUri** request AND
  - [S2] It invoked after the Client **GetStreamUri** request AND
  - [S3] RTSP address that was used to send it is equal to **trt:GetStreamUriResponse:trt:MediaUri\tt:Uri** AND
- Device response on the **RTSP DESCRIBE** request that fulfills the following requirements:
  - [S4] It has RTSP 200 response code AND
  - [S5] SDP packet contains media type "video" (m=video)
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S6] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S7] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S8] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S9] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S10] It has RTSP 200 response code AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S11] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND

- [S14] RTSP address that was used to send it is correspond to video Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to video Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

A.4 Get Audio Source Token That was Used for Streaming

Name: HelperGetAudioSourceTokenUsedForStreaming

Procedure Purpose: Get audio source token that was used in the media profile requested by the Client in **GetStreamUri** request.

Pre-requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetProfiles** operation or **AddAudioSourceConfiguration** present.

Input: GetStreamUri request

Returns: Audio Source token (*audioSourceToken*).

Annex Procedure:

- The Client Test Tool checks that there is Client **AddAudioSourceConfiguration** request or **GetProfiles** response in Test Procedure that fulfills the following requirements:

- [S1] It is invoked for the same Device as **GetStreamUri** request AND
- [S2] It is the closest one preceding **GetStreamUri** request and it fullfils [S3] or [S4] requirement AND
- If it is **AddAudioSourceConfiguration** request:
  - [S3] **trt:AddAudioSourceConfiguration/trt:ProfileToken** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
- If it is **GetProfiles** response:
  - [S4] It contains **trt:Profiles** element with **trt:Profiles/@token** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND

- The Client Test Tool checks if there is **SetAudioSourceConfiguration** command that fulfills the following requirements:
  - [S5] It invoked for the same Device as **GetStreamUri** request AND
  - If **AddAudioSourceConfiguration** request was found during previous steps:
    - [S6] It invoked after **AddAudioSourceConfiguration** request AND
    - [S7] It is the closest one preceding the **GetStreamUri** request AND
    - [S8] **trt:SetAudioSourceConfiguration/trt:Configuration/@token** value is equal to **trt:AddAudioSourceConfiguration/trt:ConfigurationToken** value AND
  - If **GetProfiles** request was found during previous steps:
    - [S9] It invoked after **GetProfiles** request AND
    - [S10] It is the closest one preceding the **GetStreamUri** request AND
    - [S11] **trt:SetAudioSourceConfiguration/trt:Configuration/@token** value is equal to **tt:AudioSourceConfiguration/@token** value from **trt:GetProfilesResponse/ trt:Profiles** with trt:Profiles/@token attribute value is equal to **trt:GetStreamUri/ trt:ProfileToken** value AND
- IF **SetAudioSourceConfiguration** command was detected during previous steps than **trt:SetAudioSourceConfiguration/trt:Configuration/tt:SourceToken** value will be returened as result of current procedure
- [S12] ELSE IF **GetProfiles** request was found during previous steps then **tt:AudioSourceConfiguration/tt:SourceToken** value from **trt:GetProfilesResponse/**

trt:Profiles element with trt:Profiles/@token is equal to **trt:GetStreamUri/trt:ProfileToken** value will be returened as result of current procedure

- ELSE IF **AddAudioSourceConfiguration** request was found during previous steps and no **SetAudioSourceConfiguration** was found during previous steps, the Client Test Tool checks the following:
  - There is **GetCompatibleAudioSourceConfigurations** request in Test Procedure that fulfills the following requirements:
    - [S13] It is invoked for the same Device as the **AddAudioSourceConfiguration** request AND
    - [S14] It is the closest one preceding the **AddAudioSourceConfiguration** request AND
    - [S15] **trt:GetCompatibleAudioSourceConfigurations/trt:ProfileToken** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
  - Device response on the **GetCompatibleAudioSourceConfigurations** request fulfills the following requirements:
    - [S16] It has HTTP 200 response code AND
    - [S17] **soapenv:Body** element has child element **trt:GetCompatibleAudioSourceConfigurationsResponse** AND
    - [S18] It contains **trt:Configuration/@token** value is equal to **trt:AddAudioSourceConfiguration/trt:ConfigurationToken** value AND
  - [S19] **trt:Configurations/tt:SourceToken** value from **trt:GetCompatibleAudioSourceConfigurationsResponse/trt:Configurations** element with @token is equal to **trt:AddAudioSourceConfiguration/trt:ConfigurationToken** value will be returened as result of current procedure

A.5 Get Audio Source Token That was Used for Streaming

Name: HelperGetAudioSourceTokenUsedForStreaming

Procedure Purpose: Get audio source token that was used in the media profile requested by the Client in **GetStreamUri** request.

Pre-requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetProfiles** operation or **AddAudioSourceConfiguration** present.

Input: GetStreamUri request

Returns: Audio Source token (*audioSourceToken*).

Annex Procedure:

- The Client Test Tool checks that there is Client **AddAudioSourceConfiguration** request or **GetProfiles** response in Test Procedure that fulfills the following requirements:
  - [S1] It is invoked for the same Device as **GetStreamUri** request AND
  - [S2] It is the closest one preceding **GetStreamUri** request and it fullfils [S3] or [S4] requirement AND
  - If it is **AddAudioSourceConfiguration** request:
    - [S3] **trt:AddAudioSourceConfiguration/trt:ProfileToken** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
  - If it is **GetProfiles** response:
    - [S4] It contains **trt:Profiles** element with **trt:Profiles/@token** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
- The Client Test Tool checks if there is **SetAudioSourceConfiguration** command that fulfills the following requirements:
  - [S5] It invoked for the same Device as **GetStreamUri** request AND
  - If **AddAudioSourceConfiguration** request was found during previous steps:
    - [S6] It invoked after **AddAudioSourceConfiguration** request AND
    - [S7] It is the closest one preceding the **GetStreamUri** request AND
    - [S8] **trt:SetAudioSourceConfiguration/trt:ConfigurationToken** value is equal to **trt:AddAudioSourceConfiguration/trt:ConfigurationToken** value AND
  - If **GetProfiles** request was found during previous steps:
    - [S9] It invoked after **GetProfiles** request AND
    - [S10] It is the closest one preceding the **GetStreamUri** request AND
    - [S11] **trt:SetAudioSourceConfiguration/trt:ConfigurationToken** value is equal to **tt:AudioSourceConfiguration** value from **trt:GetProfilesResponse/trt:Profiles** with @token attribute value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND

- IF **SetAudioSourceConfiguration** command was detected during previous steps than **trt:SetAudioSourceConfiguration/trt:Configuration/tt:SourceToken** value will be returened as result of current procedure
- [S12] ELSE IF **GetProfiles** request was found during previous steps than **tt:AudioSourceConfiguration/tt:SourceToken** value from **trt:GetProfilesResponse/ trt:Profiles** element with @token is equal to **trt:GetStreamUri/trt:ProfileToken** value will be returened as result of current procedure
- ELSE IF **AddAudioSourceConfiguration** request was found during previous steps and no **SetAudioSourceConfiguration** was found during previous steps, the Client Test Tool checks the following:
  - There is **GetCompatibleAudioSourceConfigurations** request in Test Procedure that fulfills the following requirements:
    - [S13] It is invoked for the same Device as the **AddAudioSourceConfiguration** request AND
    - [S14] It is the closest one preceding the **AddAudioSourceConfiguration** request AND
    - [S15] **trt:GetCompatibleAudioSourceConfigurations/trt:ProfileToken** value is equal to **trt:GetStreamUri/trt:ProfileToken** value AND
  - Device response on the **GetCompatibleAudioSourceConfigurations** request fulfills the following requirements:
    - [S16] It has HTTP 200 response code AND
    - [S17] **soapenv:Body** element has child element **trt:GetCompatibleAudioSourceConfigurationsResponse** AND
    - [S18] It contains **trt:Configurations/@token** value is equal to **trt:AddAudioSourceConfiguration/trt:ConfigurationToken** value AND
  - [S19] **trt:Configurations/tt:SourceToken** value from **trt:GetCompatibleAudioSourceConfigurationsResponse/trt:Configurations** element with @token is equal to **trt:AddAudioSourceConfiguration/trt:ConfigurationToken** value will be returened as result of current procedure

A.6 Find Audio Streaming corresponding to GetStreamUri

Name: HelperFindAudioStreamingForGetStreamUri

Procedure Purpose: Find audio streaming whcih corresponding to GetStreamUri pair.

Pre-requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with audio streaming present.

Input: GetStreamUri

Returns: None.

Annex Procedure:

- There is Client **RTSP DESCRIBE** request in Test Procedure that fulfills the following requirements:
  - [S1] It invoked for the same Device as **GetStreamUri** request AND
  - [S2] It invoked after the Client **GetStreamUri** request AND
  - [S3] RTSP address that was used to send it is equal to **trt:GetStreamUriResponse:trt:MediaUri\tt:Uri** AND
- Device response on the **RTSP DESCRIBE** request that fulfills the following requirements:
  - [S4] It has RTSP 200 response code AND
  - [S5] SDP packet contains media type "audio" (m=audio)
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S6] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S7] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S8] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S9] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S10] It has RTSP 200 response code AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S11] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND

- [S14] RTSP address that was used to send it is correspond to audio Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to audio Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.