ONVIF®

Profile M Client Test Specification

Profile M is under development. This test specification is not final and it is subject to changes.

Draft Version 19.12

December 2019

REVISION HISTORY

Vers.	Date	Description
19.12	Oct 22, 2019	The following was done according to #325: Check Condition based on Device Features of HTTP Digest Authentication for RTSP feauture was changed from 'Profile T' to 'Profile T or Profile M'.
19.12	Aug 23, 2019	Initial version.

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 feaures and related test cases required for testing Profile M features of a Client application e.g. metadata streaming, analytics 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 M Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile M 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 M features according to ONVIF Profile Specifications.

The principal intended purposes are:

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

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

- 3rd parties Client use cases
- 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 and HTTP protocols.

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

1.2 Test Cases for Profile Mandatory Features

This section defines test cases which are mandatory for Profile M Client conformance.

1.2.1 HTTP Digest

HTTP Digest section defines security mechanism for HTTP Digest Authentication.

1.2.2 HTTP Digest Authentication for RTSP

HTTP Digest Authentication for RTSP section defines security mechanism for Digest Authentication for RTSP.

1.2.3 Get Services

Get Services section specifies Client ability to retrieve list of services with using GetServices operation.

1.2.4 Discovery

Discovery section defines Client ability to locate services on a local network using Web Services Dynamic Discovery (WS-Discovery) protocol. It uses IP multicast address 239.255.255.250 and TCP and UDP port 3702 and SOAP-over-UDP standard for communication between nodes.

1.2.5 Device Discovery Type Filter

Device Discovery Type Filter Test Cases section defines Client ability to locate services, which are support Device Discovery Type on a local network using Web Services Dynamic Discovery (WS-Discovery) protocol. It uses IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with Types filter is equal to tds:Device or with skipped Types filter.

1.2.6 Network Configuration

Network Configuration section defines Client ability to obtain and configure of network settings on Device.

1.2.7 Metadata Streaming Using Media2 (Profile M)

Metadata Streaming Using Media2 section specifies receiving of metadata stream from Device using Madia2 Service.

1.2.8 Metadata Configuration Using Media2 (Profile M)

Metadata Configuration Using Media2 section specifies listing and modification of metadata configurations on Device.

1.3 Test Cases for Profile Conditional Features

This section defines test cases which are mandatory for Profile M Client conformance.

1.4 Test Cases for Profile Optional Features

This section defines test cases which are optional for Profile M Client conformance.

1.5 Supplementary Features and Test Cases

This section defines supplementary features and test cases which are not the part of profile, but Profile M Features results depends on them.

2 Normative References

- ONVIF Conformance Process Specification:

  https://www.onvif.org/profiles/conformance/

- ONVIF Profile Policy:

  https://www.onvif.org/profiles/

- ONVIF Network Interface Specifications:

  https://www.onvif.org/profiles/specifications/

- 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/

- ONVIF Profile M Specification:

  [TODO: put link to profile page] [http://TODO]

- 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.
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.
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.
Profile M	The Profile M Specification.
Configuration Entity	A network video device media abstract component that produces or consumes a media stream on the network, i.e. video and/or audio stream.
Media Profile	Maps a video and audio sources and outputs encoders as well as PTZ and analytics configurations.
Metadata	All streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications).

Reference Token	Token provided by the device to uniquely reference an instance of a physicalIO, configuration or profile.
Video Analytics	Algorithms or programs used to analyze video data and to generate data describing object location and behaviour.

3.3 Abbreviations

This section describes abbreviations used in this document.

HTTP	Hyper Text Transport Protocol.
HTTPS	Hyper Text Transport Protocol over Secure Socket Layer.
IP	Internet Protocol.
IPv4	Internet Protocol version 4.
TCP	Transport Control Protocol.
UDP	User Datagram Protocol.
URI	Uniform Resource Identifier.
WSDL	Web Services Description Language.
XML	eXtensible Markup Language.
RTCP	RTP Control Protocol.
RTP	Realtime Transport Protocol.
RTSP	Real Time Streaming Protocol.
SDP	Session Description Protocol.

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

Prefix	Namespace URI	Description
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
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
ter	http://www.onvif.org/ver10/error	The namespace for ONVIF defined faults
wsnt	http://docs.oasis-open.org/wsn/b-2	Schema namespace of the [WS-BaseNotification] specification.
wsa	http://www.w3.org/2005/08/addressing	Device addressing namespace as defined by [WS-Addressing].
tan	http://www.onvif.org/ver20/analytics/wsdl	The namespace for the WSDL Analytics service
tr2	http://www.onvif.org/ver20/media/wsdl	The namespace for the WSDL Media 2 service

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 conformant to Profile M is an ONVIF Client that at least supports the following features: [TODO: add breaf description of profile features as itemizedlist]

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, check condition based on Device features, required number of Devices and feature requirement level for the Profiles, which will be used for Profiles conformance.

To claim this Feature as supported Client shall pass Expected Scenario Under Test:

- for each Device, which supports Device Features defined in Check Condition Based on Device Features
- for at least with number of Devices specified in Required Number of Devices

If Feature Level Requirement is defined as Mandatory for some Profile, Client shall support this Feature to claim this Profile Conformance.

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.

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 M, 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 Test Cases for Profile Mandatory Features

5.1 HTTP Digest Test Cases

5.1.1 Feature Level Requirement:

Validated Feature: HTTP Digest authentication (HTTPDigest)

Check Condition based on Device Features: Digest

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile C Requirement: Mandatory

Profile D Requirement: Mandatory

Profile G Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile S Requirement: Mandatory

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.1.2 Expected Scenarios Under Test:

1. Client invokes a specific command which is under testing without any user credentials (no UsernameToken, no HTTP Digest authentication header).
2. Device returns HTTP 401 Unauthorized error along with WWW-Authentication: Digest header.
3. Client re-sends request with HTTP Digest Authentication header corresponding to header provided in device response.
4. Device sends a valid response to this request.
5. Client is considered as supporting HTTP Digest if the following conditions are met:
   - Device returns a valid response to specific request with HTTP Digest authentication header.

6. Client is considered as NOT supporting HTTP Digest if the following is TRUE:
   - All HTTP Digest attempts detected are failed.

5.1.3 HTTP DIGEST

Test Label: Security - HTTP Digest Authentication.

Test Case ID: HTTPDIGEST-1

Feature Under Test: HTTP Digest (HTTPDigest_HTTPDigestAuthentication)

Profile S Normative Reference: Mandatory

Profile G Normative Reference: Mandatory

Profile C Normative Reference: Mandatory

Profile Q Normative Reference: Mandatory

Profile A Normative Reference: Mandatory

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that the Client supports the HTTP Digest Authentication for HTTP level security.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with HTTP Digest Authentication present.

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

1. Client sends a request that requires authentication (e.g. GetUsers) to the Device without any authentication.
2. Device rejects the request with HTTP error code 401 AND an HTTP Digest challenge.
3. Client sends a valid request with HTTP Digest Authentication.
4. Device accepts the correct request with response code HTTP 200 OK.

Test Result:

PASS -

- [S1] Client request contains (HTTP GET method OR HTTP POST method) without any authentication AND
- Client HTTP GET request has a proper hierarchy (refer to [RFC 1945]) AND
  - [S2] Device response contains "HTTP/* 401 Unauthorized" AND
  - [S3] Device response contains "realm=*" element AND
  - [S4] Device response contains "nonce=*" element AND
  - [S5] Client request contains (HTTP GET method OR HTTP POST method) with "Authorization: Digest username=*" element AND
- Client HTTP GET request with HTTP Authentication has a proper hierarchy (refer to [RFC 1945]) AND
  - [S6] Client request contains "realm=*" element with value from Device response AND
  - [S7] Client request contains "nonce=*" element with value from Device response AND
  - [S8] Client request contains "uri=*" element AND
  - [S9] Device response contains "HTTP/* 200 OK".

FAIL -

- The Client failed PASS criteria.

5.2 HTTP Digest Authentication for RTSP Test Cases

5.2.1 Feature Level Requirement:

Validated Feature: HTTP Digest Authentication for RTSP (HTTPDigestForRTSP)

Check Condition based on Device Features: Profile T or Profile M

Required Number of Devices: 3

Profile S Requirement: None

Profile G Requirement: None

Profile A Requirement: None

Profile C Requirement: None

Profile Q Requirement: None

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.2.2 Expected Scenarios Under Test:

1. Client invokes a specific RTSP command which is under testing without any user credentials (no UsernameToken, no HTTP Digest authentication header).
2. IF Device returns HTTP 401 Unauthorized error along with WWW-Authentication: Digest header, then Client resends RTSP command with WWW-Authenticate header.
3. Client is considered as supporting HTTP Digest Authentication for RTSP if the following conditions are met:
   - Device returns a valid response to specific RTSP request with HTTP Digest authentication header.
4. Client is considered as NOT supporting HTTP Digest Authentication for RTSP if the following is TRUE:
   - All HTTP Digest attempts detected for RTSP are failed.

5.2.3 HTTP DIGEST AUTHENTICATION FOR RTSP

Test Label: HTTP Digest For RTSP

Test Case ID: HTTPDIGESTFORRTSP-1

Feature Under Test: HTTP Digest For RTSP (HTTPDigestForRTSP_HTTPDigestForRTSPTest)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile T Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Profile D Normative Reference: Conditional

Test Purpose: To verify that the Client supports the HTTP Digest Authentication for RTSP level security.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with HTTP Digest Authentication for RTSP commands present

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

1. Client sends a RTSP request that requires authentication (e.g. DESCRIBE) to the Device without any authentication.
2. Device rejects the request with a RTSP 401 status code, AND a WWW-Authenticate Response Header.
3. Client re-sends the RTSP request with a Authorization Request Header.
4. Device accepts the correct request with RTSP 200 OK status code.

Test Result:

PASS -

- There is Client RTSP request in Test Procedure that does not contain any authentication AND
- Device response on the Client RTSP request fulfills the following requirements:
  - It has RTSP 401 status code AND
  - WWW-Authenticate Response Header contains challenge = "Digest" element AND
  - WW-Authenticate Response Header contains "realm=*" element AND
  - WW-Authenticate Response Header contains "nonce=*" element AND
- There is Client RTSP request in Test Procedure that fulfills the following requirements
  - WW-Authenticate Request Header credentials = "Digest" element AND
  - WW-Authenticate Request Header contains "realm=*" element with value from Device response AND
  - WW-Authenticate Request Header contains "nonce=*" element with value from Device response AND

- WW-Authenticate Request Header contains "uri=*" element AND

- Device responds with code RTSP 200 OK.

FAIL -

- The Client failed PASS criteria.

5.3 Get Services Test Cases

5.3.1 Feature Level Requirement:

Validated Feature: Get Services (GetServices)

Check Condition based on Device Features: GetServices is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile D Requirement: Mandatory

Profile C Requirement: Mandatory

Profile G Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.3.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a services using **GetServices** commad.
2. Client is considered as supporting Get Services if the following conditions are met:
   - Client supports Capabilities_GetServicesRequest feature (please see CAPABILITIES-1 GET SERVICES section).
3. Client is considered as NOT supporting Get Services if ANY of the following is TRUE:
   - Client does not support Capabilities_GetServicesRequest feature (please see CAPABILITIES-1 GET SERVICES section).

5.4 Discovery Test Cases

5.4.1 Feature Level Requirement:

Validated Feature: Discovery (Discovery)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile S Requirement: Conditional

Profile C Requirement: Conditional

Profile G Requirement: Conditional

Profile A Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile T Requirement: Mandatory

Profile D Requirement: Mandatory

Profile M Requirement: Mandatory

5.4.2 Expected Scenarios Under Test:

1. Client sends Probe message to multicast IP address 239.255.255.250 and port 3702 to locate services on a local network.
2. Client is considered as supporting Discovery if the following conditions are met:
   - Probe request detected AND at least one ProbeMatch response detected
3. Client is considered as NOT supporting Discovery if the following is TRUE:
   - No Valid Device Response to Probe request.

5.4.3 WS-DISCOVERY

Test Label: Discovery - WS-Discovery

Test Case ID: DISCOVERY-1

Feature Under Test: WS-Discovery (Discovery_WSDiscovery)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Mandatory

Profile A Normative Reference: Mandatory

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to send Probe request and receive ProbeMatch response from Device.

Pre-Requisite:

- The Network Trace Capture files contain at least one Client Probe request to multicast IP address and one ProbeMatch response from Device directly to the Client.

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

1. Client invokes Probe request message to multicast IP address 239.255.255.250 and port 3702.
2. Device sends ProbeMatch message directly to the Client.

Test Result:

PASS -

- Client **Probe** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Probe** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<Action>" tag after the "<Header>" tag AND
  - [S2] "<Action>" includes URL address which ends with "Probe" value AND
  - [S3] Client request contains "<MessageID>" with non-empty string value AND
  - [S4] Client request contains "<Probe>" tag after the "<Body>" tag AND
  - [S5] Device response message contains "<ProbeMatches>" tag after the "<Body>" tag.

FAIL -

- The Client failed PASS criteria.

5.5 Device Discovery Type Filter Test Cases

5.5.1 Feature Level Requirement:

Validated Feature: Device Discovery Type Filter (DeviceDiscoveryTypeFilter)

Check Condition based on Device Features: Device Discovery Type is supported by Device.

Required Number of Devices: 3

Profile S Requirement: None

Profile A Requirement: Mandatory

Profile C Requirement: Conditional

Profile D Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile G Requirement: Conditional

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.5.2 Expected Scenarios Under Test:

1. Client sends Probe message to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with Types filter is equal to **tds:Device** or with skipped Types filter.
2. Client is considered as supporting Device Discovery Type if the following conditions are met:
   - **Probe** Client message that fulfills the following requirement is detected:
     - Types filter is equal to tds:Device or empty or skipped AND
     - Probe is sent to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] AND
     - Probe is sent to UDP port 3702 AND

- There is **ProbeMatch** Device message that correspond to Client **Probe**.

3. Client is considered as NOT supporting Device Discovery Type if the following is TRUE:
   - No valid Device **ProbeMatch** message that is correspond to Client **Probe** message.

5.5.3 DEVICE DISCOVERY TYPE FILTER

Test Label: Discovery - Device Discovery Type Filter

Test Case ID: DEVICEDISCOVERYTYPEFILTER-1

Feature Under Test: Device Discovery Type Filter (DeviceDiscoveryTypeFilter_DeviceDiscoveryFilter)

Profile S Normative Reference: None

Profile G Normative Reference: Mandatory

Profile C Normative Reference: Mandatory

Profile Q Normative Reference: Mandatory

Profile A Normative Reference: Mandatory

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to discover devices with Device Discovery Type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Client Probe message that does not filter out devices with Device Discovery Type that is sent to multicast WS-Discovery address.

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

1. Client invokes Probe request message to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with **Types** = tds:Device.
2. Device sends ProbeMatch message to the Client.

Test Result:

PASS -

- Client **Probe** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Probe** request in Test Procedure fulfills the following requirements:
  - [S1] It is sent to 239.255.255.250 IPv4 address OR [FF02::C] IPv6 address AND
  - [S2] It is sent to 3702 UDP port AND
  - [S3] **soapenv:Envelope/soapenv:Header** element has child element **wsadis:Action** AND
  - [S4] **wsadis:Action** includes URL address which ends with "Probe" value AND
  - [S5] **soapenv:Envelope/soapenv:Header** element has child element **wsadis:MessageID** with non-empty string value AND
  - [S6] **soapenv:Body** element has child element **d:Probe** AND
  - [S7] IF **d:Probe** element has child element **d:Types** THEN it has value is equal to **tds:Device** OR empty string value AND
  - [S8] There is Device **ProbeMatches** message in test procedure that fulfills the following requirements:
    - [S9] **soapenv:Body** element has child element **d:ProbeMatches** AND
    - [S10] **soapenv:Envelope/soapenv:Header/wsadis:RelatesTo** element value is equeal to **soapenv:Envelope/soapenv:Header/wsadis:MessageID** value in **Probe** message AND

PASS WITH WARNING -

- **d:Probe/d:Types** element is skipped OR
- **d:Probe/d:Types** element has empty string value.

FAIL -

- The Client failed PASS criteria.

5.6 Network Configuration Test Cases

5.6.1 Feature Level Requirement:

Validated Feature: Network Configuration (NetworkConfiguration)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile A Requirement: Conditional

Profile C Requirement: Conditional

Profile D Requirement: Mandatory

Profile G Requirement: Conditional

Profile Q Requirement: Conditional

Profile S Requirement: Conditional

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.6.2 Expected Scenarios Under Test:

1. Client connects to Device to configure network settings.
2. Client is considered as supporting Network Configuration if the following conditions are met:
   - Client is able to list network interfaces of Device using the GetNetworkInterfaces operation AND
   - Client is able to set network interfaces of Device using the SetNetworkInterfaces operation AND
   - Client is able to list default gateway of Device using the GetNetworkDefaultGateway operation AND
   - Client is able set default gateway of Device using the SetNetworkDefaultGateway operation.
3. Client is considered as NOT supporting Network Configuration if ANY of the following is TRUE:
   - No Valid Device Response to GetNetworkInterfaces request OR
   - No Valid Device Response to SetNetworkInterfaces request OR
   - No Valid Device Response to GetNetworkDefaultGateway request OR
   - No Valid Device Response to SetNetworkDefaultGateway request.

5.6.3 GET NETWORK INTERFACES

Test Label: Network Configuration - Get Network Interfaces

Test Case ID: NETWORKCONFIGURATION-1

Feature Under Test: Get Network Interfaces (NetworkConfiguration_GetNetworkInterfaces)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Conditional

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to list network interfaces of Device using the GetNetworkInterfaces operation.

Pre-Requisite:

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

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

1. Client invokes GetNetworkInterfaces request message to get network interface configuration from Device.
2. Device responds with code HTTP 200 OK and GetNetworkInterfacesResponse message.

Test Result:

PASS -

- Client **GetNetworkInterfaces** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetNetworkInterfaces** request in Test Procedure fulfills the following requirements:

- [S1] Client request contains "<GetNetworkInterfaces>" tag after the "<Body>" tag AND
- [S2] Device response contains "HTTP/* 200 OK" AND
- [S3] Device response contains "<GetNetworkInterfacesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

5.6.4 SET NETWORK INTERFACES

Test Label: Network Configuration - Set Network Interfaces

Test Case ID: NETWORKCONFIGURATION-2

Feature Under Test: Set Network Interfaces (NetworkConfiguration_SetNetworkInterfaces)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Conditional

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to set network interfaces of Device using the SetNetworkInterfaces operation.

Pre-Requisite:

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

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

1. Client invokes SetNetworkInterfaces request message to set the network interface configuration on Device.

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

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

5.6.5 GET NETWORK DEFAULT GATEWAY

Test Label: Network Configuration - Get Network Default Gateway

Test Case ID: NETWORKCONFIGURATION-3

Feature Under Test: Get Network Default Gateway (NetworkConfiguration_GetNetworkDefaultGateway)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Conditional

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to list default gateway of Device using the GetNetworkDefaultGateway operation.

Pre-Requisite:

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

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

1. Client invokes GetNetworkDefaultGateway request message to get the default gateway settings from Device.
2. Device responds with code HTTP 200 OK and GetNetworkDefaultGatewayResponse message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

5.6.6 SET NETWORK DEFAULT GATEWAY

Test Label: Network Configuration - Set Network Default Gateway

Test Case ID: NETWORKCONFIGURATION-4

Feature Under Test: Set Network Default Gateway (NetworkConfiguration_SetNetworkDefaultGateway)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Conditional

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to set default gateway of Device using the SetNetworkDefaultGateway operation.

Pre-Requisite:

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

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

1. Client invokes SetNetworkDefaultGateway request message to set the default gateway settings on Device.
2. Device responds with code HTTP 200 OK and SetNetworkDefaultGatewayResponse message.

Test Result:

PASS -

- Client **SetNetworkDefaultGateway** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetNetworkDefaultGateway** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<SetNetworkDefaultGateway>" tag after the "<Body>" tag AND
  - [S2] "<SetNetworkDefaultGateway>" includes tag: EITHER "<IPv4Address>" OR "<IPv6Address>" with specific IP address value AND
  - [S3] Device response contains "HTTP/* 200 OK" AND

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

FAIL -

- The Client failed PASS criteria.

5.7 Metadata Streaming Using Media2 Test Cases

5.7.1 Feature Level Normative Reference:

Validated Feature: Metadata Streaming Using Media2 for Profile M (Media2_MetadataStreaming_ProfileM)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 3

Profile M Requirement: Mandatory

5.7.2 Expected Scenarios Under Test:

1. Client connects to Device to receive metadata stream using Media2 Service.
2. Client is considered as supporting Metadata Streaming Using Media2 if the following conditions are met:
   - Client supports Media2_GetProfiles_Media2_GetProfilesRequest feature (please see MEDIA2_GETPROFILES-1 GET PROFILES USING MEDIA2 section) AND
   - Client supports Media2_GetStreamURI_Media2_GetStreamURIRequest feature (please see MEDIA2_GETSTREAMURI-1 GET STREAM URI USING MEDIA2 section) AND
   - Client supports Media2_MediaStreaming_Media2_UDP feature (please see MEDIA2_MEDIASTREAMING-2 STREAMING OVER UDP USING MEDIA2 section) OR Media2_MediaStreaming_Media2_HTTP feature (please see MEDIA2_MEDIASTREAMING-3 STREAMING OVER HTTP USING MEDIA2 section) AND
   - Client supports Media2_MetadataStreaming_MetadataStreamingUsingMedia2 feature (please see MEDIA2_METADATASTREAMING-1 METADATA STREAMING USING MEDIA2 section).
3. Client is considered as NOT supporting Metadata Streaming Using Media2 if the following is TRUE:

- Client does not support Media2_GetProfiles_Media2_GetProfilesRequest feature (please see MEDIA2_GETPROFILES-1 GET PROFILES USING MEDIA2 section) OR
- Client does not support both Media2_MediaStreaming_Media2_UDP feature (please see MEDIA2_MEDIASTREAMING-2 STREAMING OVER UDP USING MEDIA2 section) AND Media2_MediaStreaming_Media2_HTTP feature (please see MEDIA2_MEDIASTREAMING-3 STREAMING OVER HTTP USING MEDIA2 section) OR
- Client does not support Media2_GetStreamURI_Media2_GetStreamURIRequest feature (please see MEDIA2_GETSTREAMURI-1 GET STREAM URI USING MEDIA2 section) OR
- Client does not support Media2_MetadataStreaming_MetadataStreamingUsingMedia2 feature (please see MEDIA2_METADATASTREAMING-1 METADATA STREAMING USING MEDIA2 section) OR

5.8 Metadata Configuration Using Media2 Test Cases

5.8.1 Feature Level Normative Reference:

Validated Feature: Metadata Configuration Using Media2 (Media2_MetadataConfiguration_ProfileM)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 3

Profile M Requirement: Mandatory

5.8.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Metadata Configuration using Media 2 Service.
2. Client is considered as supporting Metadata Configuration Using Media2 if the following conditions are met:
   - Client supports Media2_MetadataConfiguration feature (please see Metadata Configuration Using Media2 Test Cases section).
3. Client is considered as NOT supporting Metadata Configuration Using Media2 if ANY of the following is TRUE:

- Client does not support Media2_MetadataConfiguration feature (please see Metadata Configuration Using Media2 Test Cases section).

6 Supplementary Features and Test Cases

6.1 Metadata Configuration Using Media2 Test Cases

6.1.1 Feature Level Normative Reference:

Validated Feature: Metadata Configuration Using Media2 (Media2_MetadataConfiguration)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 1

Profile T Requirement: Conditional

Profile M Requirement: Mandatory

6.1.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Metadata Configuration.
2. Client is considered as supporting Metadata Configuration if the following conditions are met:
   - Client is able to retrieve metadata configuratoins using **GetMetadataConfigurations** operation (Media2 Service) AND
   - Client is able to retrieve metadata configuratoin options using **GetMetadataConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify metadata configuratoin using **SetMetadataConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Metadata Configuration if ANY of the following is TRUE:
   - No valid response to **GetMetadataConfigurations** request (Media2 Service) OR
   - No valid response to **GetMetadataConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetMetadataConfiguration** request (Media2 Service) OR

6.1.3 GET METADATA CONFIGURATIONS USING MEDIA2

Test Label: Metadata Configuration - Get Metadata Configurations

Test Case ID: MEDIA2_METADATACONFIGURATION-1

Feature Under Test: Get Metadata Configurations Using Media2 (Media2_MetadataConfiguration_Media2_GetMetadataConfigurations)

Profile T Normative Reference: Conditional

Profile M Normative Reference: Mandatory

Test Purpose: To verify that metadata configuration provided by 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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetMetadataConfigurations** request message to retrieve an metadata configuration or a list of metadata configurations from the 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 **tr2: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 **tr2:GetMetadataConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

6.1.4 GET METADATA CONFIGURATION OPTIONS USING MEDIA2

Test Label: Metadata Configuration - Get Metadata Configuration Options

Test Case ID: MEDIA2_METADATACONFIGURATION-2

Feature Under Test: Get Metadata Configuration Options Using Media2 (Media2_MetadataConfiguration_Media2_GetMetadataConfigurationOptions)

Profile T Normative Reference: Conditional

Profile M Normative Reference: Optional

Test Purpose: To verify that metadata configuration options provided by Device is received by Client using the **GetMetadataConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetMetadataConfigurationOptions** request message to retrieve an metadata configuration options from the 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 **tr2:GetMetadataConfigurationOptions** AND

- Device response on the **GetMetadataConfigurationOptions** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetMetadataConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

6.1.5 SET METADATA CONFIGURATION USING MEDIA2

Test Label: Metadata Configuration - Set Metadata Configuration

Test Case ID: MEDIA2_METADATACONFIGURATION-3

Feature Under Test: Set Metadata Configuration Using Media2 (Media2_MetadataConfiguration_Media2_SetMetadataConfiguration)

Profile T Normative Reference: Conditional

Profile M Normative Reference: Mandatory

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetMetadataConfiguration** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **SetMetadataConfiguration** request message to change an metadata configuration on the Device.
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 **tr2:SetMetadataConfiguration** AND
- Device response on the **SetMetadataConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetMetadataConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

6.2 GET SERVICES

Test Label: Capabilities - Determine the available Services

Test Case ID: CAPABILITIES-1

Feature Under Test: Get Services (Capabilities_GetServicesRequest)

Profile S Normative Reference: Mandatory

Profile G Normative Reference: Mandatory

Profile C Normative Reference: Mandatory

Profile Q Normative Reference: Mandatory

Profile A Normative Reference: Mandatory

Profile T Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Device Capabilities is received using GetServices request.

Pre-Requisite:

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

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

1. Client invokes GetServices request message to retrieve all services of the Device.
2. Verify that GetServicesResponse message from the Device contains code HTTP 200 OK without SOAP Fault.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

6.3 METADATA STREAMING USING MEDIA2

Test Label: Metadata Streaming Using Media2

Test Case ID: MEDIA2_METADATASTREAMING-1

Feature Under Test: Metadata Streaming (Media2_MetadataStreaming_MetadataStreamingUsingMedia2)

Profile T Normative Reference: Conditional

Profile M Normative Reference: Mandatory

Test Purpose: To verify that the Client is able to retrieve the Metadata Streaming.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Metadata Streaming using Media2 Service.

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

1. Client invokes **GetStreamUri** request message for Media2 service for media profile that contains Metadata Configuration. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp 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: "application" and with encoding name "vnd.onvif.metadata" or "vnd.onvif.metadata.gzip" or "vnd.onvif.metadata.exi.onvif" or "vnd.onvif.metadata.exi.ext".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for metadata 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 or RTSP 454.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RtspOverHttp 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 "application" (m=application) with sessions attribute "rtpmap" with encoding name "vnd.onvif.metadata" OR "vnd.onvif.metadata.gzip" OR "vnd.onvif.metadata.exi.onvif" OR "vnd.onvif.metadata.exi.ext" (see ONVIF Streaming Spec) 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 invoked for Media2 Service 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 **tr2:GetStreamUriResponse\tr2: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.

6.4 STREAMING OVER UDP USING MEDIA2

Test Label: Media Streaming - UDP

Test Case ID: MEDIA2_MEDIASTREAMING-2

Feature Under Test: Streaming Over UDP Using Media2 (Media2_MediaStreaming_Media2_UDP)

Profile T Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

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:

- Device supports Media2 Real Time Streaming (Media2_RealTimeStreaming).

- 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.

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to "RtspUnicast" or "RtspMulticast".

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

1. Client invokes **GetStreamUri** request message for media profile with Transport element with "RtspUnicast" value or "RtspMulticast" value.

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.
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 or RTSP 454.

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] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
  - [S7] It has RTSP 200 response code 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] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RtspUnicast" or "RtspMulticast"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S11] It contains **tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
  - [S12] It has HTTP 200 response code 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.

6.5 STREAMING OVER HTTP USING MEDIA2

Test Label: Media Streaming - HTTP

Test Case ID: MEDIA2_MEDIASTREAMING-3

Feature Under Test: Streaming Over HTTP Using Media2 (Media2_MediaStreaming_Media2_HTTP)

Profile T Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

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:

- Device supports HTTP streaming for Media2 Service (Media2_RTPRTSPHTTP).
- 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.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to RtspOverHttp.

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

1. Client invokes **GetStreamUri** request message for media profile with Protocol element with "RtspOverHttp" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** 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 or RTSP 454.

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 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
  - [S7] It is tunneled in HTTP AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [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

- [S9] It has RTSP 200 response code 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] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RtspOverHttp"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S13] It has HTTP 200 response code AND
  - [S14] It contains **tr2: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 does not contain **Require** request header field with value is equal to "onvif-replay" AND
  - [S19] It is tunneled in HTTP 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.

6.6 GET PROFILES USING MEDIA2

Test Label: GetProfiles

Test Case ID: MEDIA2_GETPROFILES-1

Feature Under Test: Get Profiles Using Media2 (Media2_GetProfiles_Media2_GetProfilesRequest)

Profile T Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that media profiles provided by Device are 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 for Media2 Service present.

- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetProfiles** request message to retrieve a media profile or a list of media profiles from the 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] **soapenv:Body** element has child element **tr2:GetProfiles** AND
- Device response on the **GetProfiles** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetProfilesResponse**.

FAIL -

- The Client failed PASS criteria.

6.7 GET STREAM URI USING MEDIA2

Test Label: GetStreamUri

Test Case ID: MEDIA2_GETSTREAMURI-1

Feature Under Test: Get Stream URI Using Media2 (Media2_GetStreamURI_Media2_GetStreamURIRequest)

Profile T Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that stream URI provided by 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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetStreamUri** request message to retrieve a stream URI from the Device.
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] **soapenv:Body** element has child element **tr2:GetStreamUri** 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 **tr2:GetStreamUriResponse**.

FAIL -

- The Client failed PASS criteria.

Annex A Test for Appendix A

A.1 Required Number of Devices Summary

Required number of devices and Device feature dependency used in this test specification are listed in the Table.

Table A.1. Required Number of Devices Summary

Feature ID	Feature Name	Required Number of Devices	Check Condition based on Device Features	Check Condition based on Device Features ID
tc.HTTPDigest	HTTP Digest	3	Digest	Digest
tc.HTTPDigestForRTSP	HTTP Digest Authentication for RTSP	3	Profile T or Profile M	ProfileTSupported or ProfileMSupported
tc.GetServices	Get Services	3	GetServices is supported by Device.	GetServices
tc.Discovery	Discovery	3	None	All
tc.DeviceDiscoveryTypeFilter	Device Discovery Type Filter	3	Device Discovery Type is supported by Device.	DiscoveryTypesTdsDevice
tc.NetworkConfiguration	Network Configuration	3	None	All
tc.Media2_MetadataStreaming_ProfileM	Metadata Streaming Using Media2	3	Media2 Service is supported by Device.	Media2Service
tc.Media2_MetadataConfiguration_ProfileM	Metadata Configuration Using Media2	3	Media2 Service is supported by Device.	Media2Service
tc.Media2_MetadataConfiguration	Metadata Configuration Using Media2	1	Media2 Service is supported by Device.	Media2Service