

# Open Network Video Interface Forum Test Specification

Version 1.01  
September, 2009

**NOTE:**  
This ONVIF Test Specification does not cover all requirements of the ONVIF Core Specification Version 1.01

© 2009 by ONVIF: Open Network Video Interface Forum. All rights reserved.

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

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

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

## Table of Content

1	Scope.....	6
2	Normative References .....	6
3	Terms and Definitions .....	7
3.1	Definitions .....	7
3.2	Abbreviations .....	8
4	Test Overview .....	9
4.1	Basic Functionality Test.....	9
4.1.1	Device Discovery .....	9
4.1.2	Device Management.....	9
4.1.3	Media Configuration .....	11
4.1.4	Real Time Viewing .....	12
5	Test Infrastructure .....	13
5.1	Network Configuration for NVT Device .....	13
6	Test Procedure .....	14
6.1	Test Sequence .....	14
6.2	Precondition .....	15
6.3	Requirement level.....	15
6.3.1	MUST .....	15
6.3.2	MUST IF SUPPORTED .....	15
6.3.3	SHOULD, SHOULD IF SUPPORTED and OPTIONAL.....	15
7	Test Policy .....	16
7.1	IP Address Transition .....	16
7.2	Multiple Network Interfaces.....	16
7.3	Retesting.....	16
7.4	Test Logging .....	16
7.5	Device Discovery Test .....	16
7.6	Device Management Test.....	17
7.7	Media Configuration Test .....	17
7.8	Real Time Viewing Test .....	17
8	NVT Basic Functionality Test Cases.....	17
8.1	Device Discovery Test Cases.....	18
8.1.1	NVT HELLO MESSAGE.....	18
8.1.2	NVT HELLO MESSAGE VALIDATION.....	19
8.1.3	NVT SEARCH BASED ON DEVICE SCOPE TYPES.....	20
8.1.4	NVT SEARCH USING UNICAST PROBE MESSAGE.....	23
8.1.5	NVT DEVICE SCOPES CONFIGURATION.....	23
8.1.6	NVT BYE MESSAGE.....	26
8.1.7	NVT SOAP FAULT MESSAGE.....	27
8.2	Device Management Test Cases .....	28
8.2.1	NVT WSDL URL.....	28
8.2.2	NVT ALL CAPABILITIES .....	29
8.2.3	NVT DEVICE CAPABILITIES.....	30
8.2.4	NVT MEDIA CAPABILITIES.....	31
8.2.5	NVT SERVICE CATEGORY CAPABILITIES .....	32
8.2.6	NVT SOAP FAULT MESSAGE.....	32
8.2.7	NVT NETWORK COMMAND HOSTNAME CONFIGURATION.....	33

8.2.8	NVT NETWORK COMMAND DNS CONFIGURATION .....	36
8.2.9	NVT NETWORK COMMAND NTP CONFIGURATION .....	39
8.2.10	NVT SYSTEM COMMAND DEVICE INFORMATION .....	43
8.2.11	NVT SYSTEM COMMAND SYSTEMDATEANDTIME .....	44
8.2.12	NVT SYSTEM COMMAND FACTORY DEFAULT .....	50
8.2.13	NVT SYSTEM COMMAND RESET .....	52
8.3	Media Configuration Test Cases .....	54
8.3.1	NVT MEDIA PROFILE CONFIGURATION .....	54
8.3.2	NVT DYNAMIC MEDIA PROFILE CONFIGURATION .....	55
8.3.3	NVT JPEG VIDEO ENCODER CONFIGURATION .....	59
8.3.4	NVT MEDIA STREAM URI – RTP/UDP UNICAST TRANSPORT .....	60
8.3.5	NVT MEDIA STREAM URI – RTP/RTSP/HTTP TRANSPORT .....	62
8.3.6	NVT SOAP FAULT MESSAGE .....	64
8.4	Real Time Viewing Test Cases .....	66
8.4.1	NVT MEDIA CONTROL – RTSP/TCP .....	66
8.4.2	NVT MEDIA STREAMING – RTP/UDP UNICAST TRANSPORT .....	68
8.4.3	NVT MEDIA STREAMING – RTP/RTSP/HTTP TRANSPORT .....	70
8.4.4	NVT MEDIA STREAMING – RTSP KEEPALIVE .....	72
Annex A	(informative) .....	75
A.1	Invalid Device Type and Scope Type .....	75
A.2	Invalid Hostname, DNSname .....	75
A.3	Invalid Media Profile .....	75
A.4	Invalid TimeZone .....	75
A.5	Invalid RTP Header .....	75
A.6	Invalid SOAP 1.2 Fault Message .....	76
A.7	Usage of URI Life Time .....	76
A.8	Invalid WSDL URL .....	76
A.9	Valid/Invalid IPv4 Address .....	76
A.10	StreamSetup syntax for GetStreamUri .....	77

## Introduction

The goal of the ONVIF test specification is to make it possible to realize fully interoperable network video implementations from different network video vendors. The ONVIF test specification describes test framework, test infrastructure, test sequences, pre-requisites and test policies. The ONVIF test specification document refers ONVIF Core Specification v1.01 wherever necessary.

This is the ONVIF test specification. In addition, ONVIF has released the following related specifications:

- ONVIF Core Specification v1.01 [ONVIF Core]
- ONVIF Schema [ONVIF Schema]
- ONVIF Analytics Service WSDL [ONVIF Analytics WSDL]
- ONVIF Device Service WSDL [ONVIF DM WSDL]
- ONVIF Event Service WSDL [ONVIF Event WSDL]
- ONVIF Imaging Service WSDL [ONVIF Imaging WSDL]
- ONVIF Media Service WSDL [ONVIF Media WSDL]
- ONVIF PTZ Service WSDL [ONVIF PTZ WSDL]
- ONVIF Remote Discovery WSDL [ONVIF DP WSDL]
- ONVIF Topic Namespace XML [ONVIF Topic Namespace]
- ONVIF Conformance Process Specification

The purpose of this document is to define the ONVIF test framework to test Network Video Transmitter (NVT) Implementation conformance towards the ONVIF Core Specification v1.01. NVT Implementation conformance shall be validated by Network Video Client (NVC) Test Tool. NVC Test Tool is hereafter referred as “NVC”.

## Test Specification

### 1 Scope

This ONVIF Test Specification defines and regulates the conformance testing procedure for the ONVIF NVT implementation. Conformance testing is meant to be functional black-box testing. The objective of ONVIF Test Specification is to test individual requirements of NVT implementation as per ONVIF Core Specification v1.01.

The principal intended purposes are:

1. Provide self-assessment tool for implementations.
2. Provide comprehensive test suite coverage for ONVIF Core Specification v1.01.

ONVIF Test Specification does not address the following.

1. Product use cases and non-functional (performance and regression) testing.
2. SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
3. Protocol Implementation Conformance test for HTTPS, HTTP, RTP and RTSP protocols.
4. Poor streaming performance test (audio/video distortions, missing audio/video frames, incorrect lip synchronization etc)
5. Wi-Fi Conformance test

ONVIF Test Specification v1.01 will test the subset or basic functionality of the ONVIF Core Specification v1.01 and future versions of the ONVIF Test Specification will test the advanced and optional features. Refer Section 4.1 for basic functionality tests.

An NVT implementation which claims conformance to ONVIF Core Specification v1.01 MUST successfully execute all basic functionality test cases. Refer Section 8.0 for basic functionality test case descriptions.

### 2 Normative References

[ONVIF Core]	ONVIF Core Specification ver 1.01, July, 2009.
[ONVIF DM WSDL]	ONVIF Device Management Service WSDL, ver 1.0, 2009.
[ONVIF Media WSDL]	ONVIF Media Service WSDL, ver 1.0 (release candidate), 2008.
[ONVIF Schema]	ONVIF Schema, ver 1.0 (release candidate), 2008.
[ONVIF Conformance]	ONVIF Conformance Process Specification, 2008
[RFC 758]	“Assigned Numbers”, J. Postel, August 1979 <a href="http://www.ietf.org/rfc/rfc758">URL:http://www.ietf.org/rfc/rfc758</a>
[RFC 952]	“DOD INTERNET HOST TABLE SPECIFICATION”, K. Harrenstien, M. Stahl and E. Feinler, October 1985 <a href="http://www.ietf.org/rfc/rfc952">URL:http://www.ietf.org/rfc/rfc952</a>
[RFC 2119]	“Key words for use in RFCs to Indicate Requirement Levels”. S. Bradner, March 1997. <a href="http://www.ietf.org/rfc/rfc2119">URL:http://www.ietf.org/rfc/rfc2119</a>
[RFC 2131]	“Dynamic Host Configuration Protocol”, R. Droms, March 1997.

- [RFC 2136] [URL:http://www.ietf.org/rfc/rfc2131](http://www.ietf.org/rfc/rfc2131)  
“Dynamic Updates in the Domain Name System (DNS UPDATE)”, P. Vixie et. Al, April 1997.
- [RFC 2326] [URL:http://www.ietf.org/rfc/rfc2136](http://www.ietf.org/rfc/rfc2136)  
“Real Time Streaming Protocol (RTSP)”, H. Schulzrinne, A. Rao and R. Lanphier, April 1998.
- [RFC 2780] [URL:http://www.ietf.org/rfc/rfc2326](http://www.ietf.org/rfc/rfc2326)  
“IANA Allocation Guidelines For Values in the Internet”, S. Bradner and V. Paxson, March 2000
- [RFC 3550] [URL:http://www.ietf.org/rfc/rfc2780](http://www.ietf.org/rfc/rfc2780)  
“RTP: A Transport Protocol for Real-Time Applications”, H. Schulzrinne et. Al., July 2003.
- [RFC 3927] [URL:http://www.ietf.org/rfc/rfc3550](http://www.ietf.org/rfc/rfc3550)  
“Dynamic Configuration of IPv4 Link-Local Addresses”, S. Cheshire, B. Aboba and E. Guttman, May 2005.
- [RFC 3986] [URL:http://www.ietf.org/rfc/rfc3927](http://www.ietf.org/rfc/rfc3927)  
“Uniform Resource Identifier (URI): Generic Syntax”, T. Berners-Lee et. Al., January 2005.
- [RFC 4122] [URL:http://www.ietf.org/rfc/rfc3986](http://www.ietf.org/rfc/rfc3986)  
“A Universally Unique Identifier (UUID) URN Namespace”, P. Leach, M. Mealling and R. Salz, July 2005.
- [RFC 4702] [URL:http://www.ietf.org/rfc/rfc4122](http://www.ietf.org/rfc/rfc4122)  
“The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option”, M. Stapp, B. Volz and Y. Rekhter, October 2006.
- [SOAP 1.2, Part 1] [URL:http://www.ietf.org/rfc/rfc4702](http://www.ietf.org/rfc/rfc4702)  
“SOAP Version 1.2 Part 1: Messaging Framework”, M. Gudgin (Ed) et. Al, April 2007.
- [SOAP 1.2, Part 2] [URL:http://www.w3.org/TR/soap12-part1/](http://www.w3.org/TR/soap12-part1/)  
“SOAP Version 1.2 Part 2: Adjuncts (Second Edition)”, M. Gudgin (Ed) et. Al, April 2007.
- [WS-Addressing] [URL:http://www.w3.org/TR/2007/REC-soap12-part2-20070427/](http://www.w3.org/TR/2007/REC-soap12-part2-20070427/)  
“Web Services Addressing 1.0 – Core”, M. Gudgin (Ed), M. Hadley (Ed) and T. Rogers (Ed), May 2006.
- [WS-I BP 2.0] [URL:http://www.w3.org/TR/ws-addr-core/#msgaddrprops](http://www.w3.org/TR/ws-addr-core/#msgaddrprops)  
“Basic Profile Version 2.0 – Working Group Draft”, C. Ferris (Ed), A. Karmarkar (Ed) and P. Yendluri (Ed), October 2007.
- [WS-Discovery] [URL:http://www.ws-i.org/Profiles/BasicProfile-2\\_0\(WGD\).html](http://www.ws-i.org/Profiles/BasicProfile-2_0(WGD).html)  
“Web Services Dynamic Discovery (WS-Discovery)”, J. Beatty et. Al., April 2005.
- [WSDL1.1] [URL:http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf](http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf)  
“Web Services Description Language (WSDL) 1.1”, E. Christensen et. Al, March 2001.
- [XML-Schema, Part 1] [URL:http://www.w3.org/TR/wsd1](http://www.w3.org/TR/wsd1)  
“XML Schema Part 1: Structures Second Edition”, H. S. Thompson (Ed) et. Al, October 2004.
- [XML-Schema, Part 2] [URL:http://www.w3.org/TR/xmlschema-1/](http://www.w3.org/TR/xmlschema-1/)  
“XML Schema Part 2: Datatypes Second Edition”, P. V. Biron (ed) et. Al, October 2004.
- [URL:http://www.w3.org/TR/xmlschema-2/](http://www.w3.org/TR/xmlschema-2/)

### 3 Terms and Definitions

#### 3.1 Definitions

- Address** An address refers to a URI
- Capability** The capability commands allow an NVC to ask for the services provided by an NVT.
- Configuration Entity** A network video device media abstract component that is used to produce a

	media stream on the network, i.e. video and/or audio stream.
<b>Media Profile</b>	A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations.
<b>Network</b>	A network is an interconnected group of devices communicating using the Internet protocol.
<b>NVC Test Tool</b>	Network Video Client Test tool that tests the Network Video Transmitter device conformance towards [ONVIF Core]
<b>Proxy Server</b>	A server that services the requests of its clients (NVC) by forwarding requests to other servers (NVT). A Proxy provides indirect network connections to its clients (NVC).
<b>SOAP</b>	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.
<b>Switching Hub</b>	A device for connecting multiple Ethernet devices together, making them act as a single network segment.
<b>Target Service Tunneling</b>	An endpoint that makes itself available for discovery. A proxy server that passes all requests and replies unmodified.

### 3.2 Abbreviations

DUT	Device Under Test
DP	Discovery Proxy
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
HTTP	Hyper Text Transport Protocol
HTTPS	Hyper Text Transport Protocol over Secure Socket Layer
IP	Internet Protocol
IPv4	Internet Protocol version 4
JPEG	Joint Photographic Experts Group
NVT	Network Video Transmitter
NVC	Network Video Client
NTP	Network Time Protocol
POSIX	Portable Operating System Interface
PTZ	Pan/Tilt/Zoom
QVGA	Quarter Video Graphics Array
RTSP	Real Time Streaming Protocol
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
TCP	Transport Control Protocol
TTL	Time To Live
UTC	Coordinated Universal Time
USB	Universal Serial Bus
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
WSDL	Web Services Description Language
WS-I BP 2.0	Web Services Interoperability Basic Profile version 2.0
XML	eXtensible Markup Language

## 4 Test Overview

The ONVIF Test Specification v1.01 is designed to test if the device under test has implemented the basic functionality necessary to comply with [ONVIF Core]. Basic Functionality test covers basic features of NVT which includes Device Discovery, Device Management, Media Configuration and Real Time Viewing.

The future versions of ONVIF Test Specification will cover advanced and optional features of NVT i.e. Remote Discovery, Event Handling, Security, PTZ, Imaging, Analytics etc.

Refer [ONVIF Core] for the detailed description of the NVT features.

### 4.1 Basic Functionality Test

#### 4.1.1 Device Discovery

Device discovery and location of the device services in the network are achieved using a multicast discovery protocol defined in WS-Discovery. The communication between client and target service is done using Web Services, notably SOAP/UDP.

Device Discovery testing tests the following:

- Device discovery in the ad-hoc network.
- Location of one or more device services.
- Enable discovery of service by type and within scope.
- SOAP 1.2 envelopes.
- SOAP 1.2 fault messages.

Refer Table 1.0 for Device Discovery Test.

**Table 1.0 Device Discovery**

Feature	Messages
Device Discovery	Hello  Probe  Probe Match  Bye

#### 4.1.2 Device Management

Device Management defines the set of commands for retrieving device capabilities, management of network and system settings.

Device Management is done over SOAP/HTTP.

Device Management testing tests the following:

- Device Capability (all capabilities, capabilities for a particular service category).
- Device Discovery commands (scope parameter configurations).
- Network Management commands (Hostname, DNS, NTP configurations).
- System Settings (retrieve device information, configuration of system date and time, factory default reset, system reboot).
- SOAP 1.2 envelopes.
- SOAP 1.2 fault messages.

Refer Table 2.0 for Device Management Test.

**Table 2.0 Device Management**

<b>Feature</b>	<b>Commands</b>
WSDL Commands	GetWsdUrl
Capability Commands	GetCapabilities
Network Commands	GetHostname SetHostname GetDNS SetDNS GetNTP SetNTP
System Commands	GetDeviceInformation GetSystemDateAndTime SetSystemDateAndTime SetSystemFactoryDefault SystemReboot
Discovery Commands	GetScopes SetScopes AddScopes DeleteScopes

### 4.1.3 Media Configuration

Media Configuration provides streaming properties of the audio and video streams. Real time audio and video streaming configurations are controlled using a media profile. A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations.

Media Configuration (media profile and media entity) is done over SOAP/HTTP.

Media Configuration testing tests the following:

- Media Profile Configurations (retrieve existing media profiles, retrieve specific media profile).
- Creation of media profile.
- Deletion of media profile.
- Add/Remove Video Source Configurations.
- Add/Remove Video Encoder Configurations.
- Retrieve media stream URI.
- SOAP 1.2 envelopes.
- SOAP 1.2 fault messages.

Refer Table 3.0 for Media Configuration Test.

**Table 3.0 Media Configuration**

Feature	Commands
Media Profile	CreateProfile GetProfiles GetProfile AddVideoSourceConfiguration AddVideoEncoderConfiguration RemoveVideoSourceConfiguration RemoveVideoEncoderConfiguration DeleteProfile
Media Entities	GetVideoEncoderConfiguration SetVideoEncoderConfiguration

Media Stream URI	GetStreamUri
------------------	--------------

#### 4.1.4 Real Time Viewing

Real Time Viewing handles audio and video streaming and provides a mechanism for Client (NVC) to request media streams from the device under test (NVT).

Media Control is done over RTSP/TCP and media transfer over RTP/UDP.

Real Time Viewing testing tests the following:

- Real Time Streaming of audio/video streams.
- RTSP methods.
- RTP media transfer over UDP (Unicast).
- Tunneling RTP and RTSP over HTTP (firewall traversal).
- Media Streaming session liveness (RTSP Keep-alive).

Refer Table 4.0 for Real Time Viewing Test.

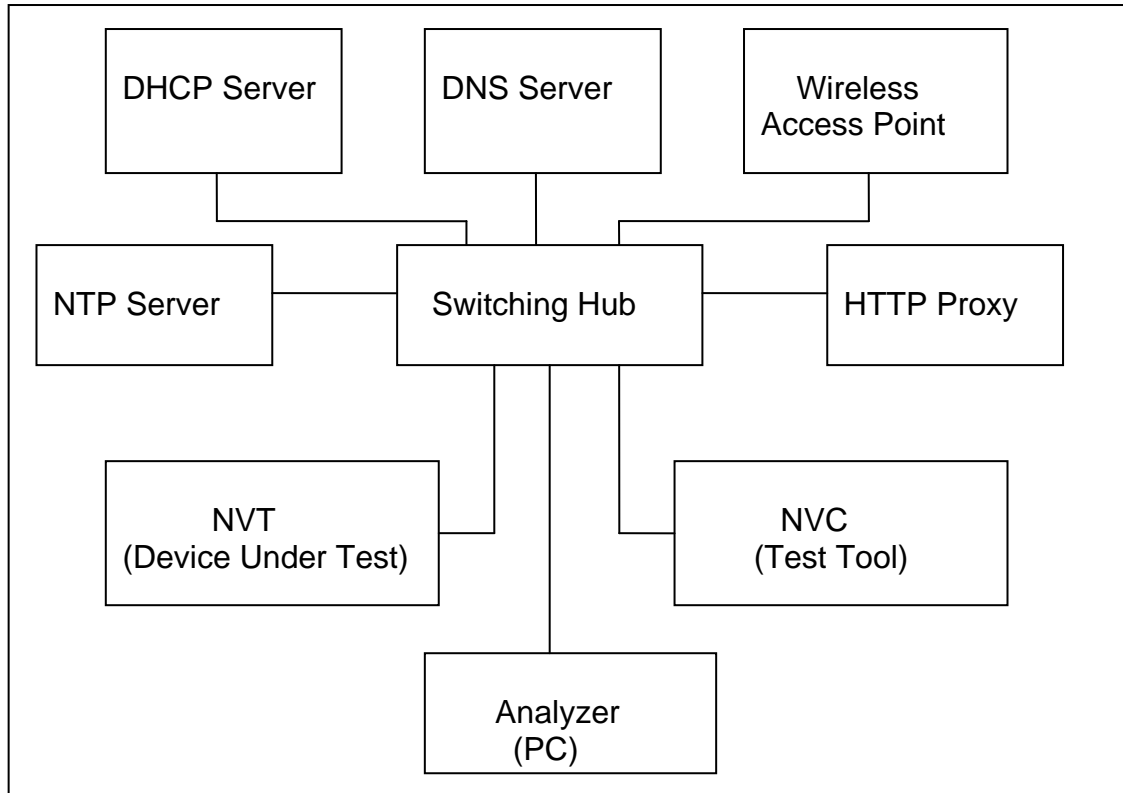
**Table 4.0 Real Time Viewing**

Feature	Functionality
Media Transport	RTP over UDP(Unicast) RTP tunneling over HTTP
Media Control	RTSP over TCP RTSP tunneling over HTTP
RTSP Method	OPTIONS DESCRIBE SETUP PLAY TEARDOWN
RTSP Keep-Alive	SET_PARAMETER (NVC->NVT)
RTP Payload Format	<b>Video Codec:</b> JPEG QVGA

## 5 Test Infrastructure

### 5.1 Network Configuration for NVT Device

Basic Functionality test cases shall be tested in the test configuration mentioned below (figure 1.0).



**Figure 1.0 Test Configuration for NVT Device**

**NVT:** Device under test

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

**HTTP Proxy:** facilities in RTP and RTSP tunneling over HTTP.

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

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

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

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

**Analyzer PC:** capture the packets in real time and save the packet capture log information of the failure test cases.

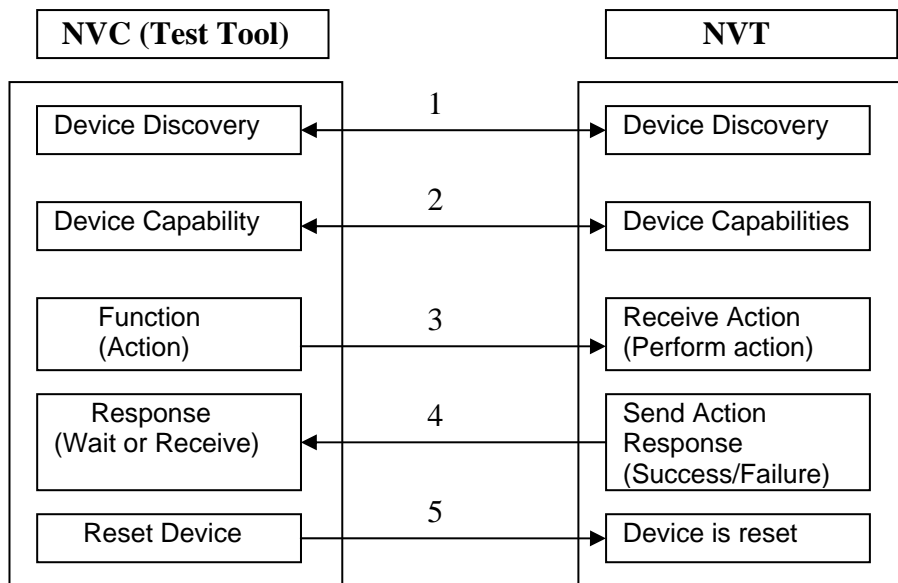
**Switching Hub:** All devices should be connected to the Hub.

## 6 Test Procedure

### 6.1 Test Sequence

This section describes the generic test sequence between NVC Test Tool and NVT. All tests are executed by the NVC. NVC will test for both success and failure test scenarios.

This generic test sequence diagram (figure 2.0) is to be considered as an example. All basic functionality test cases are illustrated with test sequence diagram. Refer Section 8.0 for basic functionality test cases.



**Figure 2.0 Generic Test Sequence Diagram**

#### 1. Device Discovery

- a. NVC will discover the DUT.
- b. NVC will locate the Device Services on the network.

#### 2. Device Capability

- a. NVC will retrieve DUT capabilities.
- b. DUT respond with its capabilities.

#### 3. Action Request

- a. NVC will perform the required action on the DUT.
- b. DUT will perform the action.

#### 4. Action Response

- a. NVC will wait and receive the action response from the DUT.

- b. DUT will send action response (success/failure).

## 5. Device Reset

- a. NVC will reset the DUT to the original state.
- b. DUT state is reset.

## 6.2 Precondition

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

- The DUT must be configured with an IPv4 address.
- The DUT must be IP reachable [in the test configuration].
- The DUT must be configured with the time i.e. manual configuration of UTC time and if NTP is supported by DUT then NTP time must be synchronized with NTP Server.

## 6.3 Requirement level

The general interpretation of the requirement levels is as defined in [RFC2119]. The following sections describe how the requirement levels affect the test procedure.

### 6.3.1 MUST

Test cases that cover parts of [ONVIF Core] that are mandatory to implement in all ONVIF conformant products have the requirement level "MUST". The test result for these test cases MUST be "PASSED" for the DUT to be ONVIF conformant.

### 6.3.2 MUST IF SUPPORTED

The requirement level "MUST IF SUPPORTED" is used for test cases that cover parts of [ONVIF Core] that are mandatory to implement if and only if the DUT supports the referenced service, feature or functional block in any possible way.

If the DUT does support the referenced service, feature or functional block, then the test result MUST be "PASSED" for the DUT to be ONVIF conformant.

If the DUT does not support the referenced service, feature or functional block, then the DUT MUST correctly reply with a proper fault message to be ONVIF conformant. The test result in this case MUST be "DEVICE FEATURE NOT SUPPORTED BY NVT".

### 6.3.3 SHOULD, SHOULD IF SUPPORTED and OPTIONAL

The "SHOULD" level indicates that the service, functional block or feature, SHOULD be implemented by the DUT. The "SHOULD IF SUPPORTED" level indicates that the service, functional block or feature, SHOULD be implemented by the DUT if supported by the DUT in any way. The "OPTIONAL" level indicates that the service, functional block or feature, MAY or MAY NOT be implemented by the DUT. Failure to comply with these requirement levels is not a violation of the ONVIF Conformance requirement. However, if the ONVIF support is implemented, then it MUST be done in conformance with [ONVIF Core].

If the referenced part of [ONVIF Core] has been implemented in the DUT, then the test result MUST be “PASSED” for the DUT to be ONVIF conformant.

If the referenced part of [ONVIF Core] has not been implemented in the DUT, then the test should not be executed.

## 7 Test Policy

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

### 7.1 IP Address Transition

IPv4 address of DUT and NVC are configured by one of the following means.

- Static IPv4 Address
- DHCP based

During the testing, IP address change or address transition is not permitted. If this happens, then all testing shall be repeated from the beginning i.e. Device Discovery Test.

### 7.2 Multiple Network Interfaces

A device under test that has multiple network interfaces (Wired Ethernet i.e. 802.3af and Wireless Ethernet i.e. 802.11a/b/g/n), initial testing will be performed on the Wired Ethernet network interface. After completion of all testing on the Wired Ethernet network interface, all tests shall be repeated on Wireless Ethernet network interface.

ONVIF Test Specification restricts all testing to Wired Ethernet and/or Wireless Ethernet network interface, other interfaces like USB, Bluetooth etc are outside the scope of the testing.

### 7.3 Retesting

At any time during the testing, the DUT may enter into an unrecoverable state (e.g. a system crash or a hung) and NVC is no longer able to perform the prescribed test procedure, then the DUT will be rebooted and the test shall be restarted from the beginning i.e. Device Discovery Test.

### 7.4 Test Logging

All test sequences are analyzed by the packet capture tool (ex: Wire Shark running on a PC). If device under test exhibits a failure condition, the packet capture log shall be saved for further analysis. Packet capture will be stopped and restarted at multiple times throughout the test procedure.

### 7.5 Device Discovery Test

- The device under test must be discovered by the NVC device that exists in the testing environment.
- Failure to discover the device on the network constitutes failure of the test procedure.
- Failure to locate the device services on the network constitutes failure of the test procedure.

- Failure to select the device for interaction constitutes failure of the test procedure.
- Certain DUT's may not support device discovery feature, in such situations, device discovery tests shall not be executed and NVC will directly communicate with the DUT. Discovery mode settings of the DUT can be retrieved through **GetDiscoveryMode** SOAP command.

Refer Section 8.1 for Device Discovery Test Cases.

## 7.6 Device Management Test

- The device under test must demonstrate the Device and Media capabilities. A NVT that does not list the mandatory device capability constitutes failure of the test procedure.
- If DUT does not support NTP Configuration commands (i.e. Get NTP Settings and Set NTP Settings) then it MUST respond to the request with SOAP 1.2 fault message (ActionNotSupported).

Refer Section 8.2 for Device Management Test Cases.

## 7.7 Media Configuration Test

- The device under test must support at-least one media profile with Video Configuration. Video Configuration must include video source and video encoder media entities.
- The device under test must support JPEG QVGA video encoding.
- In certain test cases, NVC may create new media configuration (i.e. media profile and media entities). In such cases, the test procedure will delete those modified configuration at the end of the test procedure.

Refer Section 8.3 for Media Configuration Test Cases.

## 7.8 Real Time Viewing Test

- Media Control Stream URI shall be retrieved by **GetStreamUri** SOAP command.
- NVC and DUT time should be synchronized for media streaming.
- For real time media contents, NVT must be able to serve the content and NVC must be able to receive media streams at a rate sufficient for real time rendering (streaming).
- Inability to stream full length media content (i.e. start to end) by NVT constitutes failure of the test procedure.
- Real Time Viewing testing will only test one media stream at a time.
- Poor streaming performance test is outside the scope of the ONVIF Test Specification.

Refer Section 8.3 for Real Time Viewing Test Cases.

## 8 NVT Basic Functionality Test Cases

This section describes the test procedure for basic functionality test cases.

## 8.1 Device Discovery Test Cases

This section covers tests designed for NVT Device Discovery Feature.

### 8.1.1 NVT HELLO MESSAGE

**Test Label:** Device Discovery NVT Multicast HELLO Message Transmission.

**ONVIF Core Specification Coverage:** 7.3.3 Hello

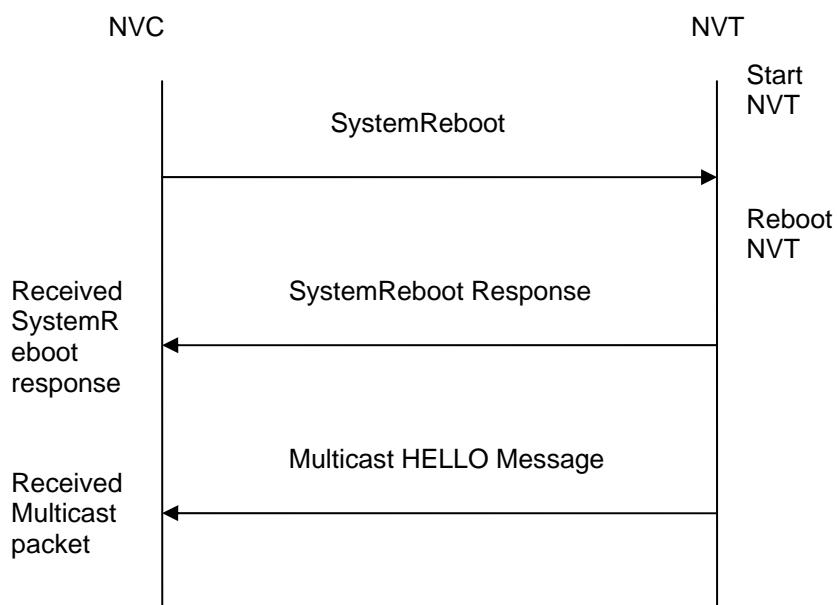
**Device Type:** NVT

**Requirement Level:** MUST

**Test Purpose:** To verify that the NVT transmits HELLO message with the correct multicast parameters (address, port number and TTL) when it is connected to the network.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC invokes SystemReboot message to reboot the NVT.

4. NVT sends SystemRebootResponse message.

5. NVC waits for the user defined boot time to receive HELLO message from NVT.

6. Verify that the NVT transmits the HELLO message with multicast address **239.255.255.250**, port number **3702** and TTL of **1**.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SystemRebootResponse message.

The DUT did not send the multicast HELLO message.

### 8.1.2 NVT HELLO MESSAGE VALIDATION

**Test Label:** Device Discovery NVT HELLO Message Validation

**ONVIF Core Specification Coverage:** 7.3.1 Endpoint reference, 7.3.3.1 Types, 7.3.3.2 Scopes

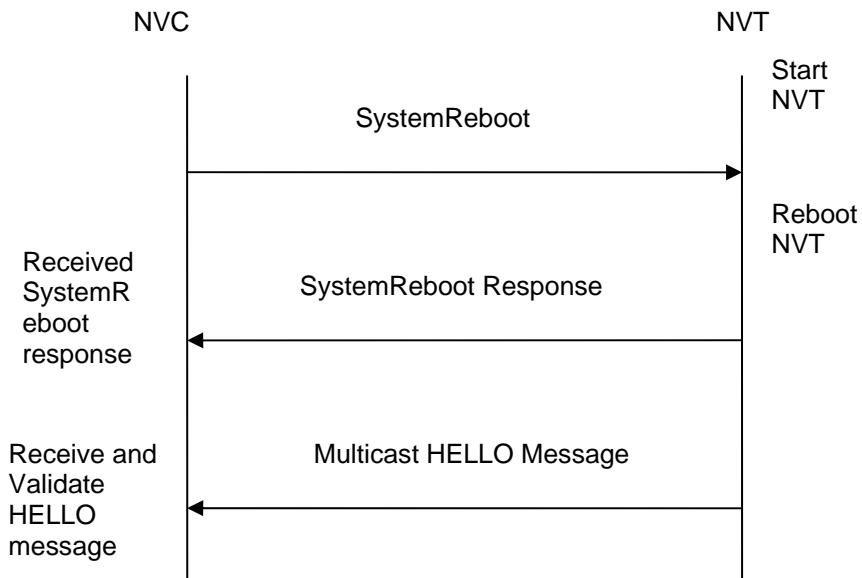
**Device Type:** NVT

**Requirement Level:** MUST

**Test Purpose:** To verify the mandatory XML elements Device type, Scope types, Endpoint Reference and Meta data version in the HELLO message.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC invokes SystemReboot message to reboot the NVT.

4. NVT sends SystemRebootResponse message.

5. NVC waits for the user defined boot time to receive HELLO message from NVT.

6. NVC will verify the mandatory XML elements in the NVT HELLO message.

**Test Result: PASS** – DUT passes all assertions

**FAIL** – The DUT did not send SystemRebootResponse message.

The DUT did not send multicast HELLO message.

The DUT did not send HELLO message with one or more mandatory XML elements.

The DUT did not send HELLO message with mandatory device type and scope types (type, location, hardware and name).

**Note:** See Annex A for Device and Scope Types definition.

### 8.1.3 NVT SEARCH BASED ON DEVICE SCOPE TYPES

**Test Label:** Device Discovery NVT Search based on device scope types.

**ONVIF Core Specification Coverage:** 7.3.3.2 Scopes, 7.3.4 Probe and Probe Match

**Device Type:** NVT

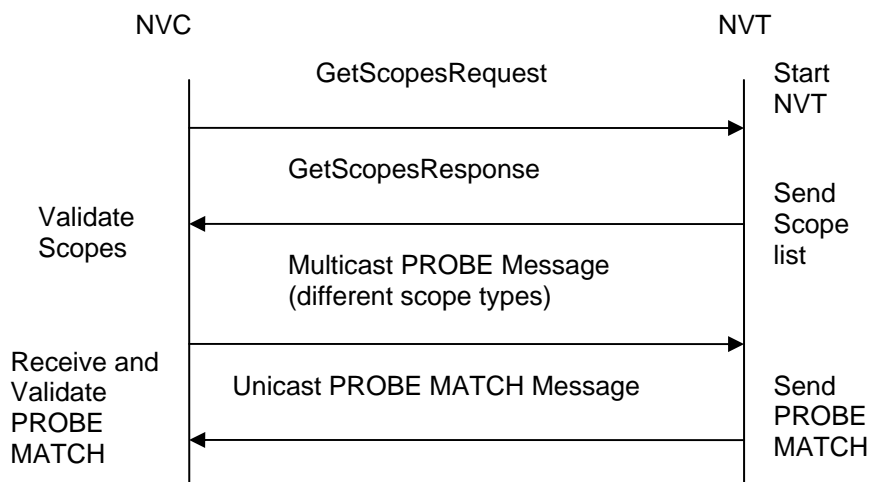
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To search the NVT based on the mandatory scope types (type, location, hardware and name).

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.
3. NVC will invoke GetScopesRequest message to retrieve existing scopes list.
4. NVT replies with the list of scopes types in GetScopesResponse message.
5. NVC will transmit the multicast PROBE message with different scope types (type, location, hardware and name).
6. NVC will verify the PROBE MATCH message sent by NVT.

**Test Result: PASS** – DUT passes all assertions

**FAIL** – The DUT did not send GetScopesResponse message.

The DUT scope list does not have one or more mandatory scope entry.

The DUT did not send PROBE MATCH message between 0 to 500 milliseconds.

The DUT did not send PROBE MATCH message.

The DUT did not send mandatory XML elements (device, scope type, service address and scope matching rule) in the PROBE MATCH message.

**Note:** See Annex A for Device and Scope Types definition.

#### 8.1.3.1 NVT SEARCH WITH OMITTED DEVICE AND SCOPE TYPES

**Test Label:** Device Discovery NVT Search with omitted device type and scope types.

**ONVIF Core Specification Coverage:** 7.3.3.2 Scopes, 7.3.4 Probe and Probe Match

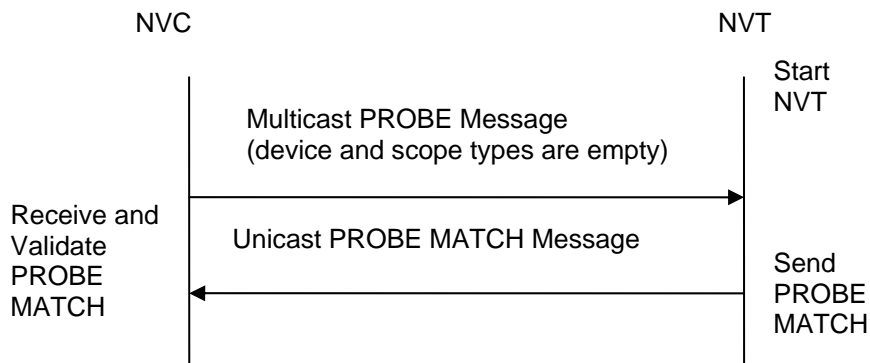
**Device Type:** NVT

**Requirement Level:** MUST

**Test Purpose:** To search the NVT with device and scope types being omitted.

**Test Configuration:** NVC and NVT

**Test Sequence:**



- Test Procedure:**
1. Start an NVC.
  2. Start an NVT.
  3. NVC will transmit multicast PROBE message with device type and scope type inputs omitted.
  4. NVC will verify the PROBE MATCH message sent by NVT.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send PROBE MATCH message between 0 to 500 milliseconds.

The DUT did not send PROBE MATCH message.

The DUT did not send mandatory XML elements (device, scope type, service address and scope matching rule) in the PROBE MATCH message.

### 8.1.3.2 NVT RESPONSE TO INVALID SEARCH REQUEST

**Test Label:** Device Discovery NVT does not respond to invalid multicast PROBE message.

**ONVIF Core Specification Coverage:** 7.3.4 Probe and Probe Match

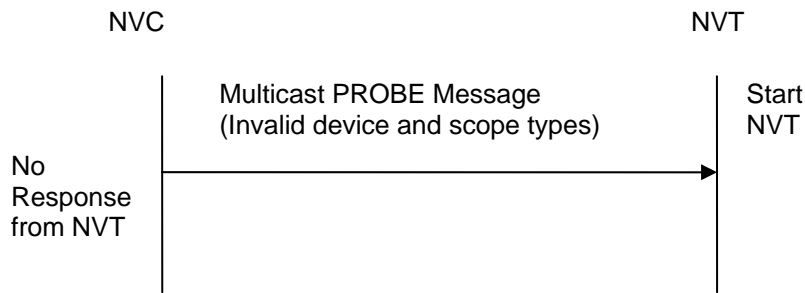
**Device Type:** NVT

**Requirement Level:** MUST

**Test Purpose:** To verify that NVT do not reply to the invalid multicast PROBE message (invalid device and scope types).

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will transmit multicast PROBE message with invalid device and scope types.

4. Verify that the NVT did not send PROBE MATCH message.

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did send PROBE MATCH message.

**Note:** See Annex A for Invalid Device and Scope Types definition.

#### 8.1.4 NVT SEARCH USING UNICAST PROBE MESSAGE

**Note:** All Tests 8.1.3, 8.1.3.1, 8.1.3.2 to be repeated with Unicast PROBE message.

#### 8.1.5 NVT DEVICE SCOPES CONFIGURATION

**Test Label:** Device Discovery NVT Device Scope configurations.

**ONVIF Core Specification Coverage:** 8.3.11 Get scope parameters, 8.3.12 Set scope parameters, 8.3.13 Add scope parameters, 8.3.14 Delete scope parameters

**Device Type:** NVT

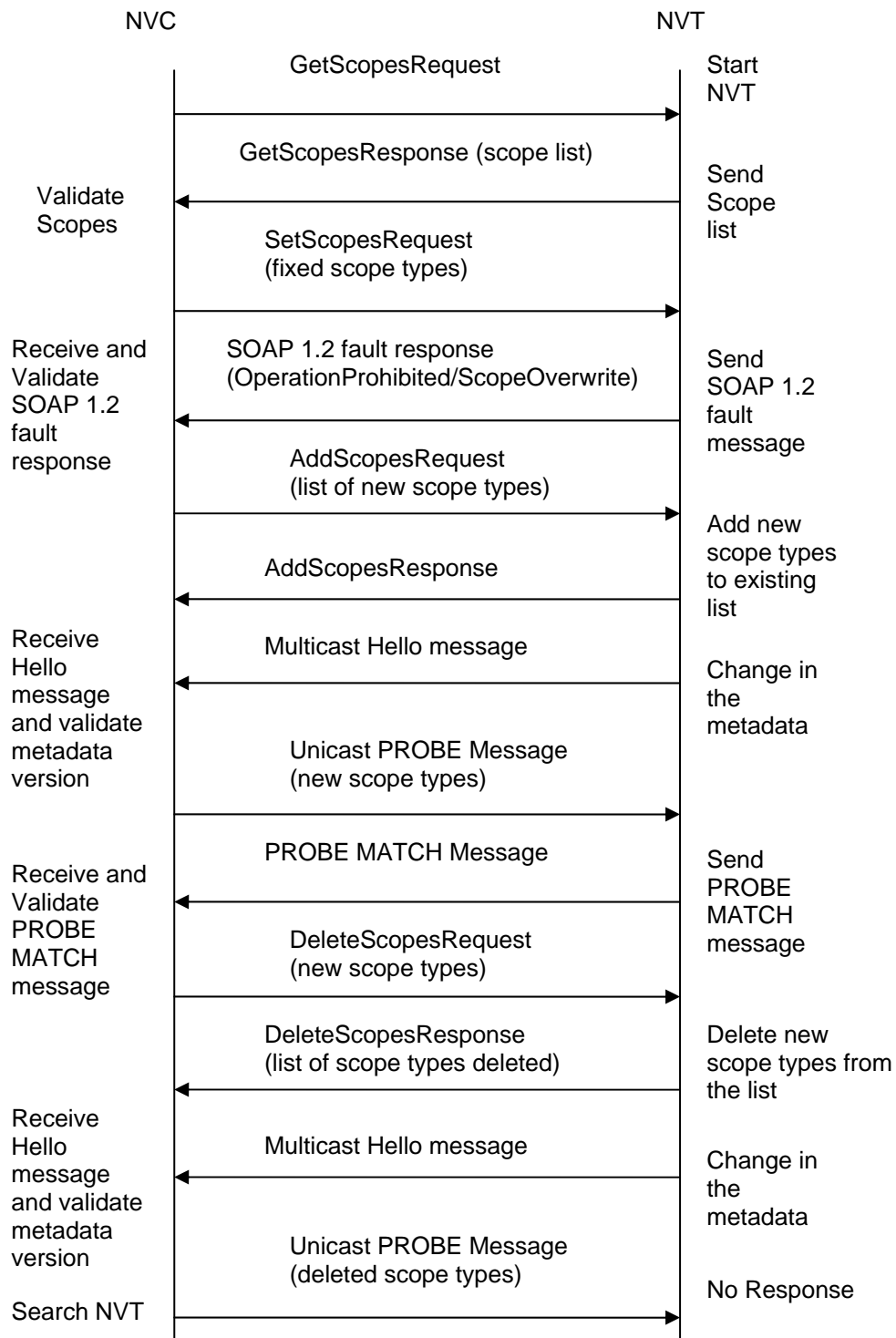
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify NVT behaviour for scope parameter configuration.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetScopesRequest message to retrieve existing scope types.

4. NVT replies with the list of scopes types in the GetScopesResponse message.

5. NVC will invoke SetScopesRequest message to overwrite existing scope types with new scope types.

6. NVT generates SOAP 1.2 fault message as fixed scope types cannot be overwritten.

7. NVC will invoke AddScopesRequest message to add new scope types to the existing scope list.

8. NVT replies with AddScopesResponse message indicating success.

9. NVT sends Multicast Hello message to indicate the change in the metadata (i.e. addition of new scope types to the existing list).

10. NVC will invoke Unicast PROBE message to search NVT with newly added scope types.

11. Verify that NVT issued a PROBE MATCH message.

12. NVC will invoke DeleteScopesRequest message to delete the newly configured scope types.

13. NVT replies with DeleteScopesResponse message indicating success.

14. NVT sends Multicast Hello message to indicate the change in the metadata (i.e. deletion of scope types from the existing list).

15. NVC will invoke Unicast PROBE message to search NVT with deleted scope types.

16. Verify that the NVT did not send PROBE MATCH message.

**Test Result: PASS** – DUT passes all assertions

**FAIL** – The DUT did not send GetScopesResponse message.

The DUT scope list does not have one or more mandatory scope entry.

The DUT did not send SOAP 1.2 fault message (OperationProhibited/ScopeOverwrite).

The DUT did not send AddScopesResponse message.

The DUT did not send multicast Hello message after the change in its metadata (addition/deletion of scope types).

The DUT did not send PROBE MATCH message between 0 to 500 milliseconds.

The DUT did not send PROBE MATCH message.

The DUT did not send mandatory XML elements (device, new scope type, service address and scope matching rule) in the PROBE MATCH message.

The DUT did not send DeleteScopesResponse message.

**Note:** Whenever there is a change in the metadata of the Target Service, “**MetadataVersion**” is incremented by  $\geq 1$ .

See Annex A for Invalid SOAP 1.2 fault message definition.

### 8.1.6 NVT BYE MESSAGE

**Test Label:** Device Discovery NVT BYE Message Transmission.

**ONVIF Core Specification Coverage:** 7.3.6 Bye

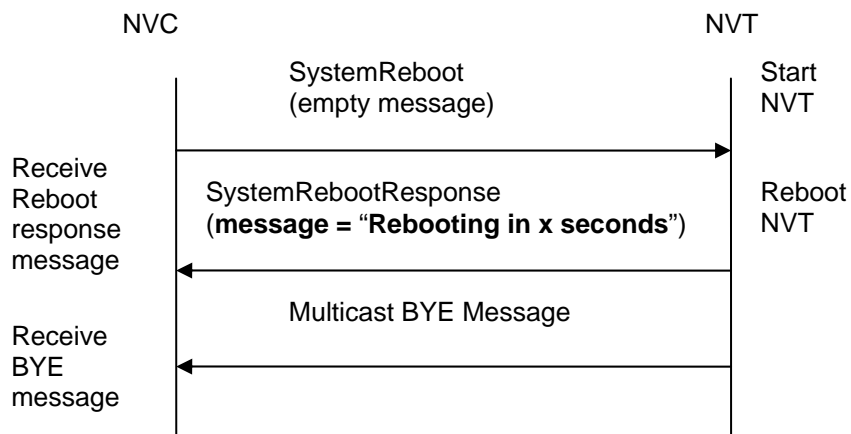
**Device Type:** NVT

**Requirement Level:** SHOULD

**Test Purpose:** To verify that NVT transmits BYE message before the system reboot.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SystemReboot message to reboot the NVT.

4. Verify that NVT sends SystemRebootResponse message (example **message** string = “**Rebooting in x seconds**”).

5. Verify that the NVT issued a BYE message.

6. NVC waits for the user defined boot time before proceeding to execute next test case.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SystemRebootResponse message.

The DUT did not send BYE message.

### 8.1.7 NVT SOAP FAULT MESSAGE

**Test Label:** Device Discovery NVT generates SOAP 1.2 fault message for Invalid Unicast PROBE Message.

**ONVIF Core Specification Coverage:** 7.3.7 SOAP Fault Messages

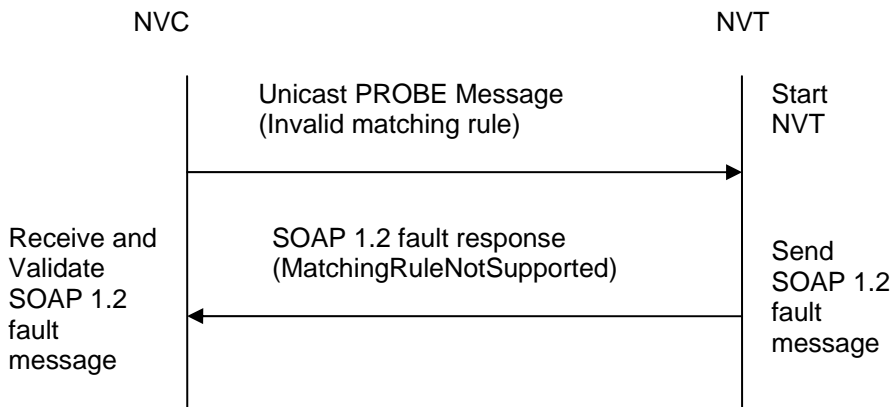
**Device Type:** NVT

**Requirement Level:** OPTIONAL

**Test Purpose:** To verify that NVT generates a SOAP 1.2 fault message to the invalid Unicast PROBE message (Invalid matching rule).

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will transmit Unicast PROBE message with invalid matching type rule.

4. Verify that the NVT generates a SOAP 1.2 fault message (MatchingRuleNotSupported).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

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

**Note:** See Annex A for Invalid SOAP 1.2 fault message definition. Refer RFC 3986 for scope matching definitions.

## 8.2 Device Management Test Cases

This section covers tests designed for NVT Device Management Feature.

### 8.2.1 NVT WSDL URL

**Test Label:** Device Management NVT WSDL URL.

**Command under test:** GetWsdUrl

**ONVIF Core Specification Coverage:** 8.1.1 Get WSDL URL

**Device Type:** NVT

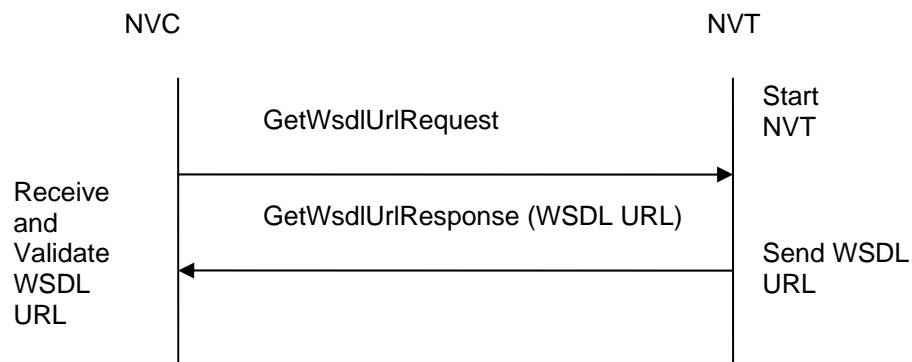
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To retrieve complete XML schema and WSDL definitions of the NVT.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetWsdUrlRequest message to retrieve XML schema and WSDL definitions of the NVT.

4. Verify that NVT sends GetWsdUrlResponse message (WSDL URL).

5. Validate the WSDL URL returned from the NVT.

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetWsdUrlResponse message.

The DUT did not send correct WSDL URL i.e. incorrectly formed.

**Note:** See Annex A for Invalid WSDL URL definition.

### 8.2.2 NVT ALL CAPABILITIES

**Test Label:** Device Management NVT Device Capabilities Verification.

**Command under test:** GetCapabilities

**ONVIF Core Specification Coverage:** 8.1.2 Capability exchange

**Device Type:** NVT

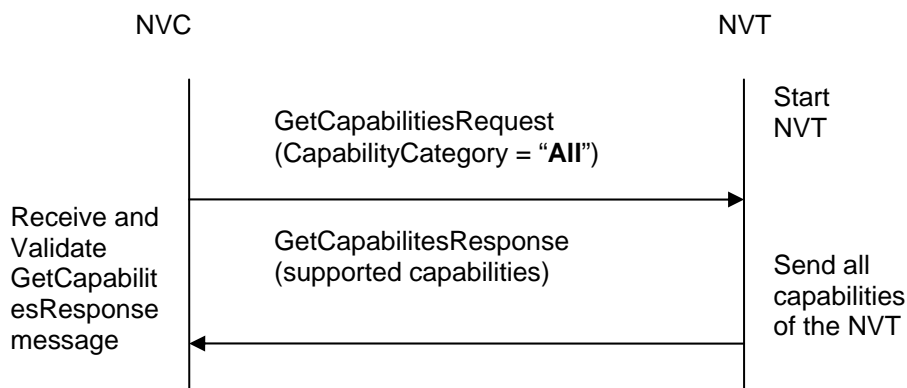
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify all Capabilities of the NVT.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetCapabilitiesRequest message (CapabilityCategory = "All") to retrieve all capabilities of the NVT.

4. Verify the Capabilities Response from the NVT and support for Device and Media capabilities.

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetCapabilitesResponse message.

The DUT did not support Device and Media capabilities.

### 8.2.3 NVT DEVICE CAPABILITIES

**Test Label:** Device Management NVT Device Capabilities Verification.

**Command under test:** GetCapabilities

**ONVIF Core Specification Coverage:** 8.1.2 Capability exchange

**Device Type:** NVT

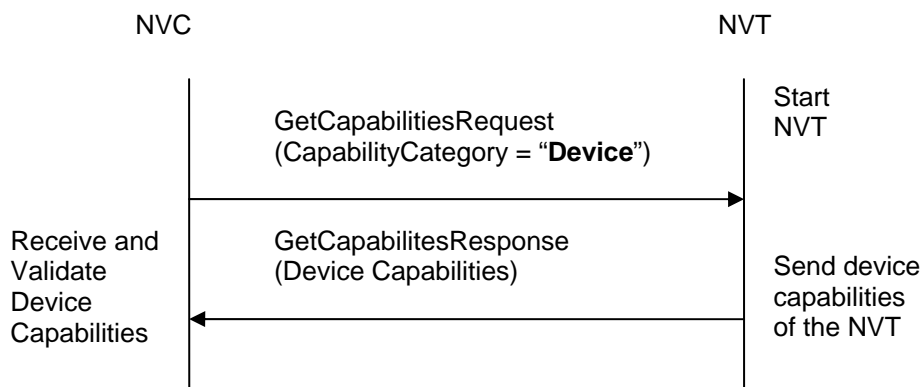
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify Device Capabilities of the NVT.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetCapabilitiesRequest message (CapabilityCategory = "Device") to retrieve Device Capabilities of the NVT.

4. NVT sends its device capabilities in the GetCapabilitesResponse message.

5. Verify the address of the device service in the GetCapabilitesResponse message.

6. Verify Network, System, IO and Security capabilities if supported by the DUT.

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetCapabilitesResponse message.

The DUT did not send the address of the device service.

The DUT did not send correct ONVIF version i.e. 1.0.

### 8.2.4 NVT MEDIA CAPABILITIES

**Test Label:** Device Management NVT Media Capabilities Verification.

**Command under test:** GetCapabilities

**ONVIF Core Specification Coverage:** 8.1.2 Capability exchange

**Device Type:** NVT

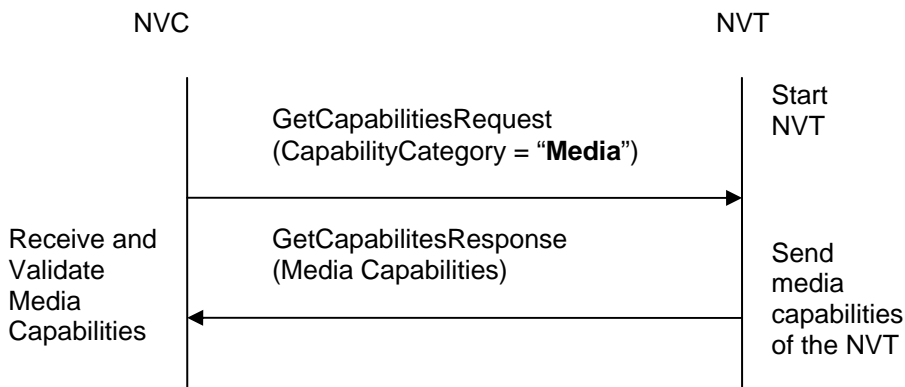
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify Media Capabilities of the NVT.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetCapabilitiesRequest message (CapabilityCategory = "Media") to retrieve Media Capabilities of the NVT.

4. NVT sends its media capabilities in the GetCapabilitesResponse message.

5. Verify the address of the media service in the GetCapabilitesResponse message.

6. Verify Real time streaming capabilities if supported by the DUT.

**Test Result:** PASS – DUT passes all assertions.

**FAIL** – The DUT did not send GetCapabilitesResponse message.

The DUT did not send the address of the media service.

### 8.2.5 NVT SERVICE CATEGORY CAPABILITIES

**Note:** NVT Service Category Capabilities Test to be repeated for each service category like Analytics, Events, Imaging and PTZ.

If a specific service category is not supported by the DUT, it MUST generate SOAP 1.2 fault response (ActionNotSupported/NoSuchService).

### 8.2.6 NVT SOAP FAULT MESSAGE

**Test Label:** Device Management NVT generates a SOAP 1.2 fault message for Invalid GetCapabilitiesRequest Message.

**Command under test:** GetCapabilities

**ONVIF Core Specification Coverage:** 8.1.2 Capability exchange

**Device Type:** NVT

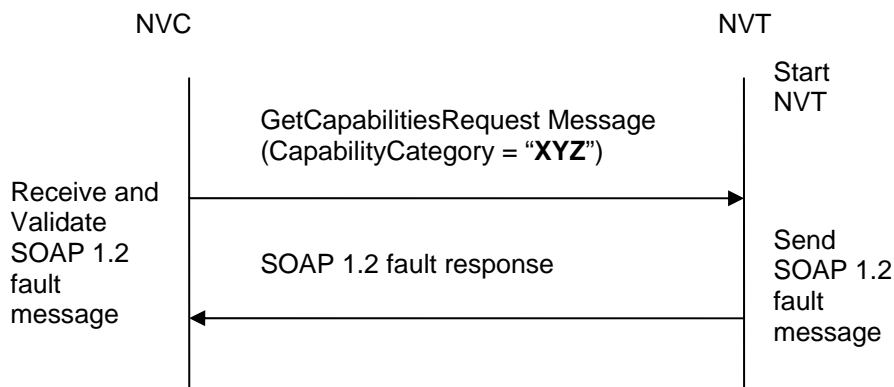
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify that NVT generates a SOAP 1.2 fault message to the invalid GetCapabilitiesRequest message (invalid capability category).

**Test Configuration:** NVC and NVT

**Test Sequence:**



- Test Procedure:**
1. Start an NVC.
  2. Start an NVT.
  3. NVC will send GetCapabilitiesRequest message with invalid capability category.
  4. Verify that the NVT generates a SOAP 1.2 fault message.

**Test Result:** PASS – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

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

**Note:** See Annex A for Invalid SOAP 1.2 fault message definition.

### 8.2.7 NVT NETWORK COMMAND HOSTNAME CONFIGURATION

**Test Label:** Device Management NVT Network Command **GetHostname** Test.

**Command Under Test:** GetHostname

**ONVIF Core Specification Coverage:** 8.2.1 Get hostname

**Device Type:** NVT

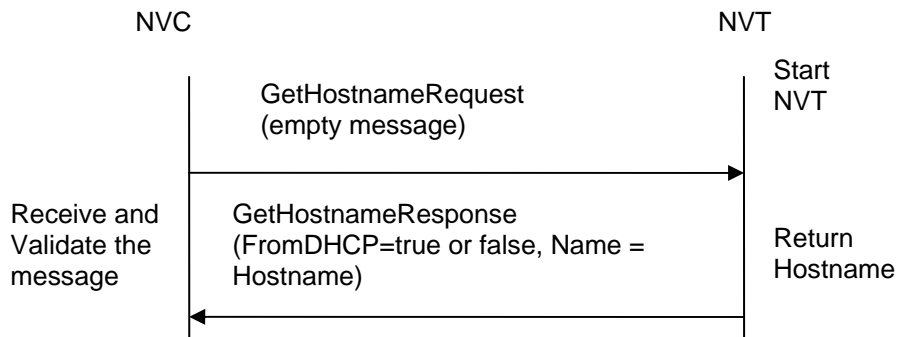
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To retrieve hostname of the NVT through **GetHostname** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetHostnameRequest message to retrieve Hostname of the NVT.

4. Verify the GetHostnameResponse from NVT (FromDHCP = true or false, Name = Hostname).

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetHostnameResponse message.

### 8.2.7.1 NVT NETWORK COMMAND SETHOSTNAME TEST

**Test Label:** Device Management NVT Network Command **SetHostname** Test.

**Command Under Test:** SetHostname

**ONVIF Core Specification Coverage:** 8.2.2 Set hostname

**Device Type:** NVT

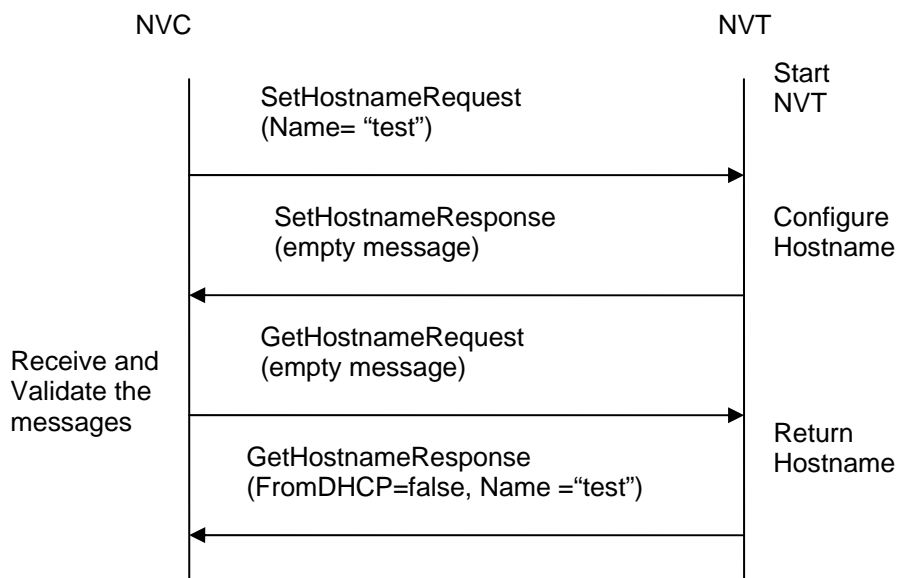
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To configure hostname on an NVT through **SetHostname** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetHostnameRequest message (Name = **“test”**) to configure the hostname.

4. Verify that the NVT sends SetHostnameResponse (empty message).

5. Verify the hostname configurations in NVT through GetHostnameRequest.

6. NVT sends hostname configuration in the GetHostnameResponse message (FromDHCP = false, Name = **“test”**).

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SetHostnameResponse message.

The DUT did not send GetHostnameResponse message.

The DUT did not send correct hostname (i.e. “**test**”) in the GetHostnameResponse message.

**8.2.7.2 NVT NETWORK COMMAND SETHOSTNAME TEST**

**Test Label:** Device Management NVT Network Command **SetHostname** Test.

**Command Under Test:** SetHostName

**ONVIF Core Specification Coverage:** 8.2.2 Set hostname

**Device Type:** NVT

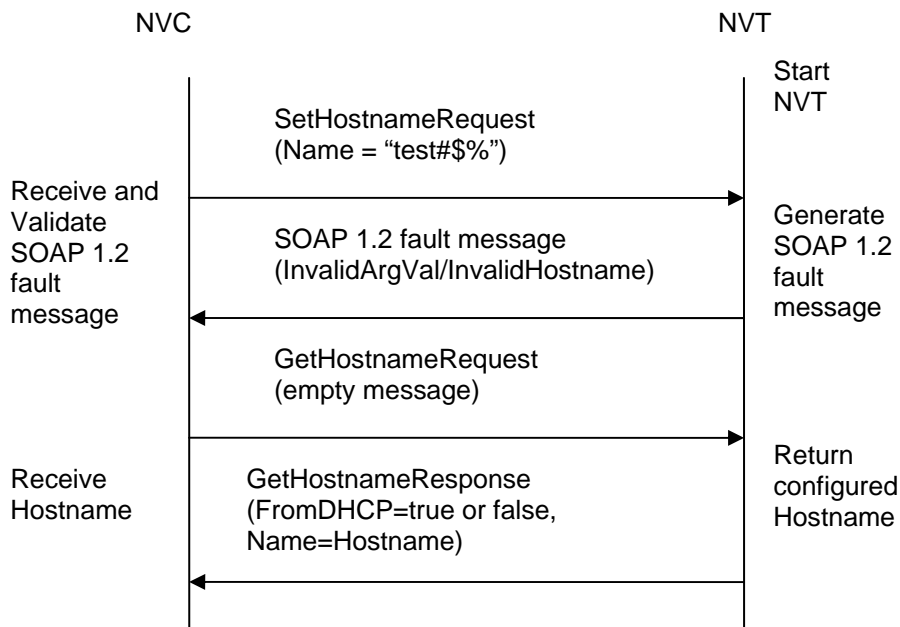
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify behaviour of NVT for invalid hostname configuration.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetHostnameRequest message (Name = “**test#\$%**”) to configure the hostname.

4. Verify that the NVT generates SOAP 1.2 fault message (InvalidArgVal/InvalidHostname).
5. Verify hostname from NVT through GetHostnameRequest.
6. NVT sends valid hostname in GetHostnameResponse message (FromDHCP=true or false, Name=Hostname).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct fault code in the SOAP fault message (InvalidArgVal/InvalidHostname).

The DUT did not send GetHostnameResponse message.

The DUT returned “test#\$\$” as its Hostname.

**Note:** Hostname “test#\$\$” is just an example. See Annex A for Invalid Hostname and SOAP 1.2 fault message definitions.

### 8.2.8 NVT NETWORK COMMAND DNS CONFIGURATION

**Test Label:** Device Management NVT Network Command **GetDNS** Test.

**Command Under Test:** GetDNS

**ONVIF Core Specification Coverage:** 8.2.3 Get DNS settings

**Device Type:** NVT

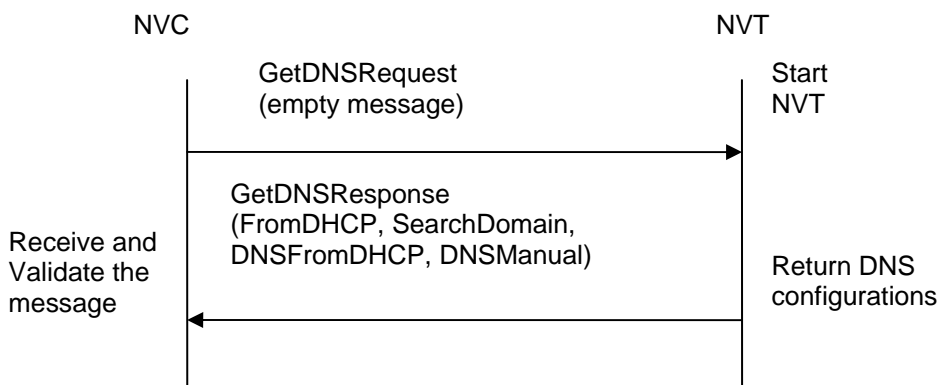
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To retrieve DNS configurations of NVT through **GetDNS** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetDNSRequest message to retrieve DNS configurations of the NVT.

4. Verify the GetDNSResponse from NVT (FromDHCP = true or false, SearchDomain = domain to search if hostname is not fully qualified, DNSFromDHCP = list of DNS Servers obtained from DHCP, DNSManual = list of DNS Servers manually configured).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetDNSResponse message.

### 8.2.8.1 NVT NETWORK COMMAND SETDNS TEST

**Test Label:** Device Management NVT Network Command **SetDNS** Test.

**Command Under Test:** SetDNS

**ONVIF Core Specification Coverage:** 8.2.4 Set DNS settings

**Device Type:** NVT

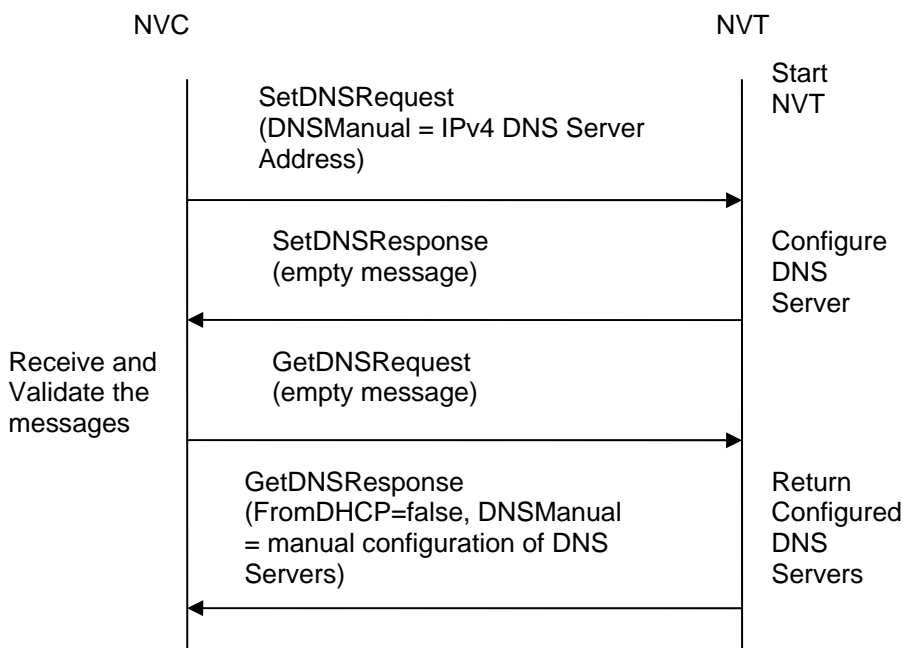
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To configure DNS settings in NVT through **SetDNS** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



- Test Procedure:**
1. Start an NVC.
  2. Start an NVT.
  3. NVC will invoke SetDNSRequest message (DNSManual = "IPv4", "10.1.1.1").
  4. Verify that the NVT sends SetDNSResponse (empty message).
  5. Verify the DNS configurations in NVT through GetDNSRequest.
  6. NVT sends its DNS configurations in the GetDNSResponse message (FromDHCP = **false**, DNSManual = "IPv4", "10.1.1.1").

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SetDNSResponse message.

The DUT did not send GetDNSResponse message.

The DUT did not send correct information (i.e. FromDHCP=**false**, DNSManual = "IPv4", "10.1.1.1") in the GetDNSResponse message.

**Note:** See Annex A for Valid IPv4 Address definition.

#### 8.2.8.2 NVT NETWORK COMMAND SETDNS TEST

**Test Label:** Device Management NVT Network Command **SetDNS** Test.

**Command Under Test:** SetDNS

**ONVIF Core Specification Coverage:** 8.2.4 Set DNS settings

**Device Type:** NVT

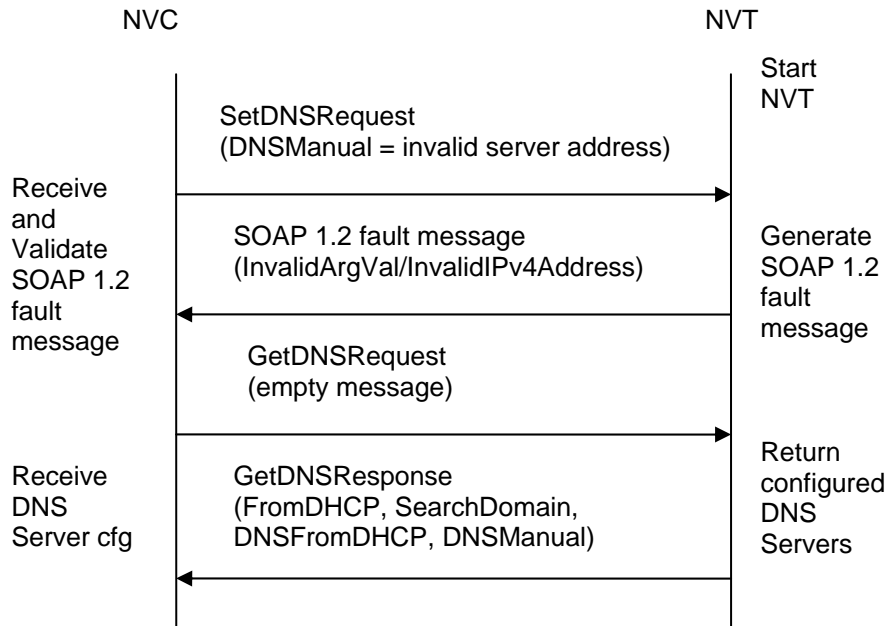
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify behaviour of NVT for invalid DNS Server Address Configuration.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetDNSRequest message (DNSManual = "IPv4", "10.1.1.255").

4. Verify that the NVT generates SOAP 1.2 fault message (InvalidArgVal/InvalidIPv4Address).

5. Retrieve DNS configurations from NVT through GetDNSRequest.

6. NVT sends valid DNS configurations in the GetDNSResponse message (FromDHCP=true or false, SearchDomain = domain to search if hostname is not fully qualified, DNSFromDHCP = list of DNS Servers obtained from DHCP, DNSManual = list of manual DNS Servers).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct fault code in the SOAP fault message (InvalidArgVal/InvalidIPv4Address).

The DUT did not GetDNSResponse message.

The DUT returned "10.1.1.255" as DNS Server address.

**Note:** See Annex A for Invalid IPv4 Address and SOAP 1.2 fault message definitions.

## 8.2.9 NVT NETWORK COMMAND NTP CONFIGURATION

**Test Label:** Device Management NVT Network Command **GetNTP** Test.

**Command Under Test:** GetNTP

**ONVIF Core Specification Coverage:** 8.2.5 Get NTP settings

**Device Type:** NVT

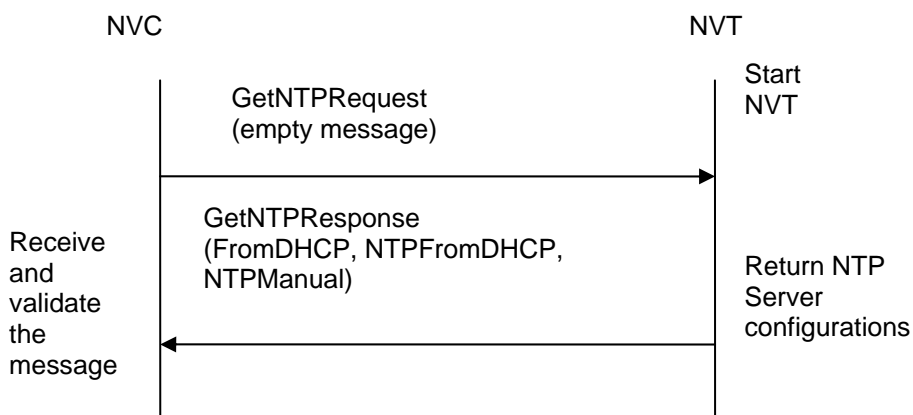
**Requirement Level:** MUST IF SUPPORTED

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To retrieve NTP Server settings of the NVT through **GetNTP** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetNTPRequest message to retrieve NTP Server settings of the NVT.

4. Verify the GetNTPResponse from NVT (FromDHCP = true or false, NTPFromDHCP = list of NTP Servers obtained from DHCP, NTPManual = list of NTP Servers manually configured).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetNTPResponse message.

**Note:** If DUT does not support NTP Get Configuration command, then it MUST respond to a request with SOAP 1.2 fault message (ActionNotSupported).

### 8.2.9.1 NVT NETWORK COMMAND SETNTP TEST

**Test Label:** Device Management NVT Network Command **SetNTP** Test.

**Command Under Test:** SetNTP

**ONVIF Core Specification Coverage:** 8.2.6 Set NTP settings

**Device Type:** NVT

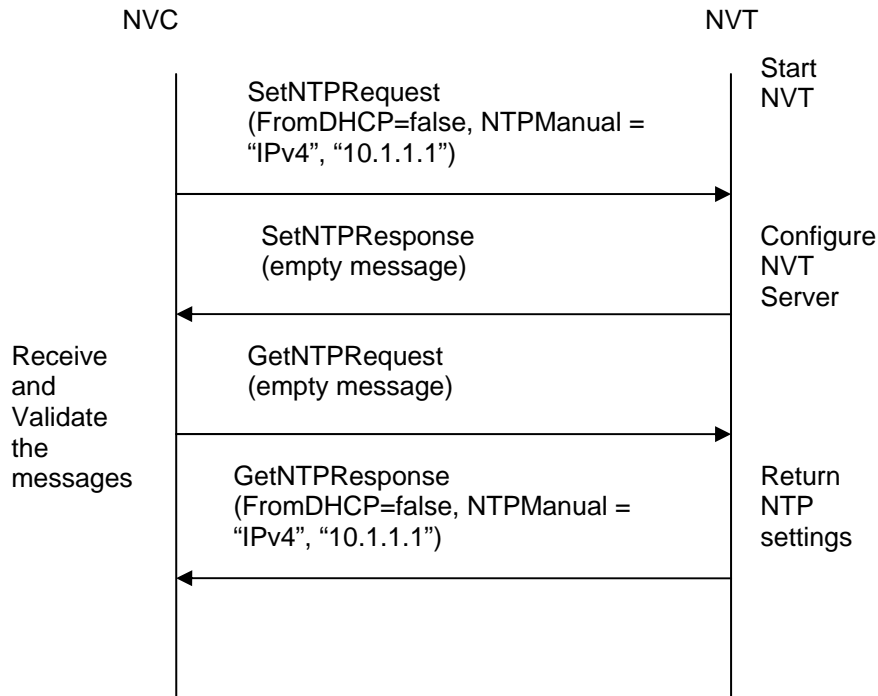
**Requirement Level:** MUST IF SUPPORTED

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To configure NTP settings on an NVT through **SetNTP** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke **SetNTPRequest** message (**FromDHCP = false**, **NTPManual [Type = "IPv4", IPv4Address = "10.1.1.1"]**).

4. Verify that the NVT sends **SetNTPResponse** (empty message).

5. Verify the NTP Server settings in NVT through **GetNTPRequest** message.

6. NVT sends its NTP Server settings in the **GetNTPResponse** message (**FromDHCP= false**, **NTPManual [Type = "IPv4", IPv4Address = "10.1.1.1"]**).

**Test Result:** **PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send **SetNTPResponse** message in step-4.

The DUT did not send GetNTPResponse message in step-6.

The DUT did not send correct NTP Server information (i.e. FromDHCP=**false**, NTPManual = "IPv4", "10.1.1.1") in GetNTPResponse message in step-6..

**Note:** If DUT does not support NTP Set Configuration command, then it **MUST** respond to a request with SOAP 1.2 fault message (ActionNotSupported).

See Annex A for Valid IPv4 Address definition.

### 8.2.9.2 NVT NETWORK COMMAND SETNTP TEST

**Test Label:** Device Management NVT Network Command **SetNTP** Test.

**Command Under Test:** SetNTP

**ONVIF Core Specification Coverage:** 8.2.6 Set NTP settings

**Device Type:** NVT

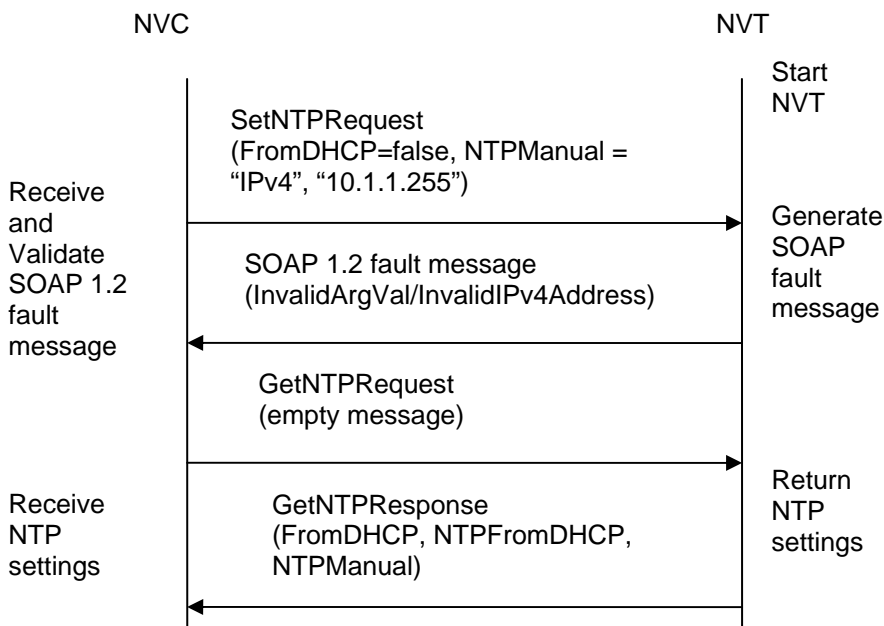
**Requirement Level:** MUST IF SUPPORTED

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify behaviour of NVT for Invalid IPv4 address configuration.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetNTPRequest message (FromDHCP = false, NTPManual [Type = "IPv4", IPv4Address = "10.1.1.255"]).

4. Verify that the NVT generates SOAP 1.2 fault message (InvalidArgVal/InvalidIPv4Address).

5. Retrieve NTP Server configurations from NVT through GetNTPRequest message.

6. NVT sends valid NTP Server configurations in the GetNTPResponse message (FromDHCP = true or false, NTPFromDHCP = list of NTP Servers obtained from DHCP, NTPManual = list of NTP Servers manually configured).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct fault code in the SOAP fault message (InvalidArgVal/InvalidIPv4Address).

The DUT did not GetNTPResponse message.

The DUT returned "10.1.1.255" as NTP Server address.

**Note:** If DUT does not support NTP Set Configuration command, then it MUST respond to a request with SOAP 1.2 fault message (ActionNotSupported).

See Annex A for Invalid IPv4 Address and SOAP 1.2 fault message definitions.

## 8.2.10 NVT SYSTEM COMMAND DEVICE INFORMATION

**Test Label:** Device Management NVT System Command **GetDeviceInformation** Test.

**Command Under Test:** GetDeviceInformation

**ONVIF Core Specification Coverage:** 8.3.1 Device Information

**Device Type:** NVT

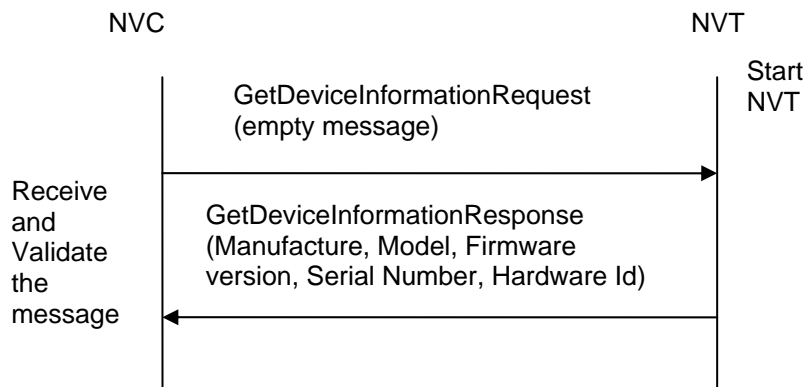
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To retrieve device information of NVT through **GetDeviceInformation** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetDeviceInformationRequest message to retrieve device information such as manufacture, model and firmware version etc.

4. Verify the GetDeviceInformationResponse from NVT (Manufacture, Model, Firmware version, Serial Number and Hardware Id).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetDeviceInformationResponse message.

The DUT did not send one or more mandatory information in the GetDeviceInformationResponse message (mandatory information - Manufacture, Model, Firmware Version, Serial Number and Hardware Id)

### 8.2.11 NVT SYSTEM COMMAND SYSTEMDATEANDTIME

**Test Label:** Device Management NVT System Command **GetSystemDateAndTime** Test.

**Command Under Test:** GetSystemDateAndTime

**ONVIF Core Specification Coverage:** 8.3.4 Get system date and time

**Device Type:** NVT

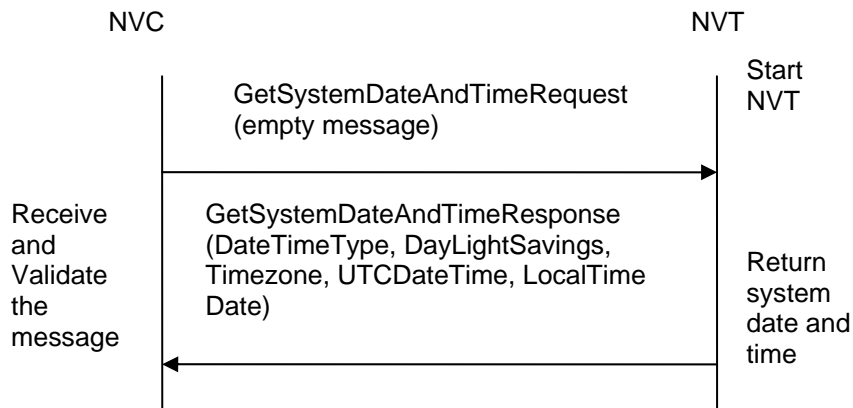
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To retrieve NVT system date and time through **GetSystemDateAndTime** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetSystemDateAndTimeRequest message to get NVT system date and time.

4. Verify system date and time configurations of NVT in GetSystemDateAndTimeResponse message (DateTimeType = Manual or NTP, DayLightSavings = true or false, Timezone = POSIX 1003.1, UTC DateTime = Hour:Min:Sec, Year:Month:Day and LocalTimeDate = Hour:Min:Sec, Year:Month:Day).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send GetSystemDateAndTimeResponse message.

The DUT did not send DateTimeType and DayLightSavings information in the GetSystemDateAndTimeResponse message.

**Note:** If system date and time are set manually, then DUT MUST return UTCDateTime or LocalDateTime in the GetSystemDateAndTimeResponse message.

### 8.2.11.1 NVT SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST

**Test Label:** Device Management NVT Network Command **SetSystemDateAndTime** Test.

**Command Under Test:** SetSystemDateAndTime

**ONVIF Core Specification Coverage:** 8.3.5 Set system date and time

**Device Type:** NVT

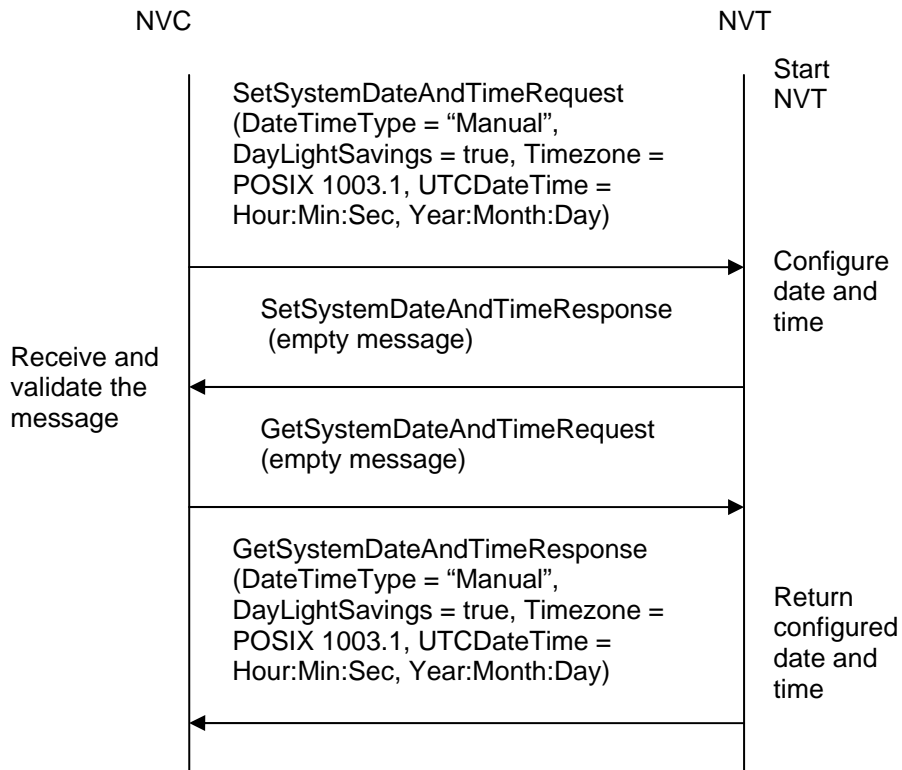
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To set the NVT system date and time through **SetSystemDateAndTime** command.

**Test Configuration: NVC and NVT**

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetSystemDateAndTimeRequest message (DateTimeType = **Manual**, DayLightSavings = **true**, Timezone = **POSIX 1003.1**, UTCDateTime = **Hour:Min:Sec, Year:Month:Day**).

4. Verify that the NVT sends SetSystemDateAndTimeResponse (empty message).

5. Verify the NVT date and time configurations through GetSystemDateAndTimeRequest message.

6. NVT sends system date and time configurations in the GetSystemDateAndTimeResponse message (DateTimeType = **Manual**, DayLightSavings = **true**, Timezone = **POSIX 1003.1**, UTCDateTime = **Hour:Min:Sec, Year:Month:Day**).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SetSystemDateAndTimeResponse message.

The DUT did not send GetSystemDateAndTimeResponse message.

The DUT did not send expected system date and time configuration (DateTimeType = "Manual", DayLightSavings = true, Timezone = **POSIX 1003.1**, UTCDateTime = **Hour:Min:Sec, Year:Month:Day**) in the GetSystemDateAndTimeResponse message.

### 8.2.11.2 NVT SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST

**Test Label:** Device Management NVT Network Command **SetSystemDateAndTime** Test.

**Command Under Test:** SetSystemDateAndTime

**ONVIF Core Specification Coverage:** 8.3.5 Set system date and time

**Device Type:** NVT

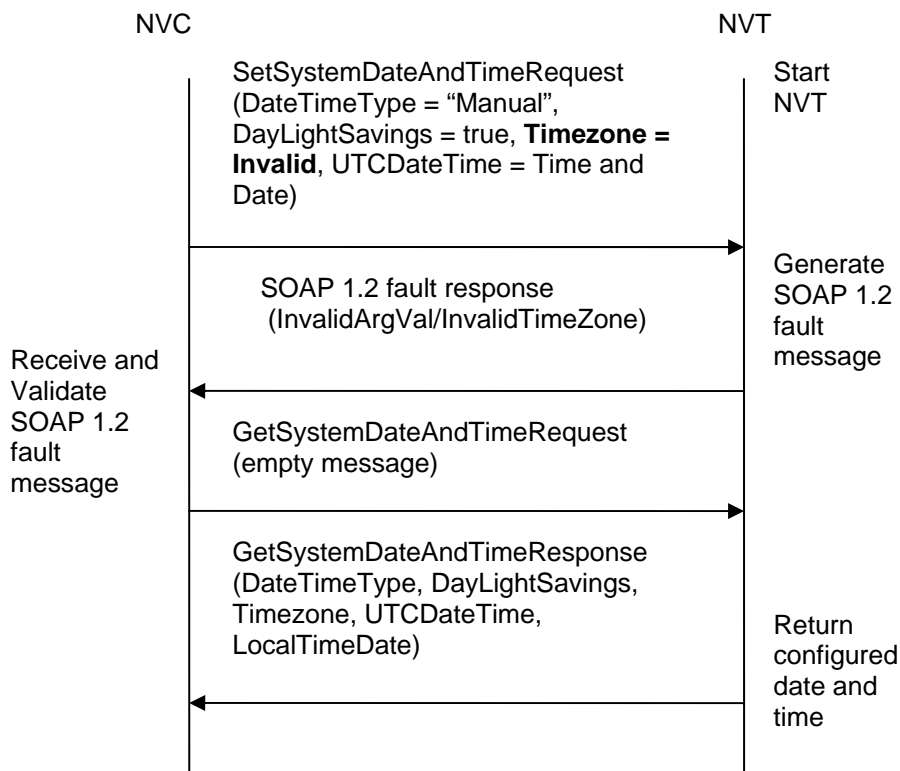
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify the behaviour of NVT for invalid Timezone configuration.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.
3. NVC will invoke SetSystemDateAndTimeRequest message with invalid Timezone (DateTimeType = "Manual", DayLightSavings = on, **Timezone = Invalid**, UTCDateTime = Hour:Min:Sec, Year:Month:Day).
4. Verify that NVT generates SOAP 1.2 fault response (InvalidArgVal/InvalidTimezone).
5. Verify the NVT system date and time configurations through GetSystemDateAndTimeRequest message.
4. NVT sends system date and time configurations in the GetSystemDateAndTimeResponse message (DateTimeType = Manual or NTP, DayLightSavings = true or false, Timezone = POSIX 1003.1, UTC DateTime = Hour:Min:Sec, Year:Month:Day and LocalTimeDate = Hour:Min:Sec, Year:Month:Day).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct fault code in the SOAP fault message (InvalidArgVal/InvalidTimezone).

The DUT did not GetSystemDateAndTimeResponse message.

The DUT returned "**Invalid Timezone**" in the GetSystemDateAndTimeResponse message.

**Note:** If system date and time are set manually, then DUT MUST return UTCDateTime or LocalDateTime in the GetSystemDateAndTimeResponse message.

See Annex A for Invalid Timezone and SOAP 1.2 fault message definitions.

### 8.2.11.3 NVT SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST

**Test Label:** Device Management NVT Network Command **SetSystemDateAndTime** Test.

**Command Under Test:** SetSystemDateAndTime

**ONVIF Core Specification Coverage:** 8.3.5 Set system date and time

**Device Type:** NVT

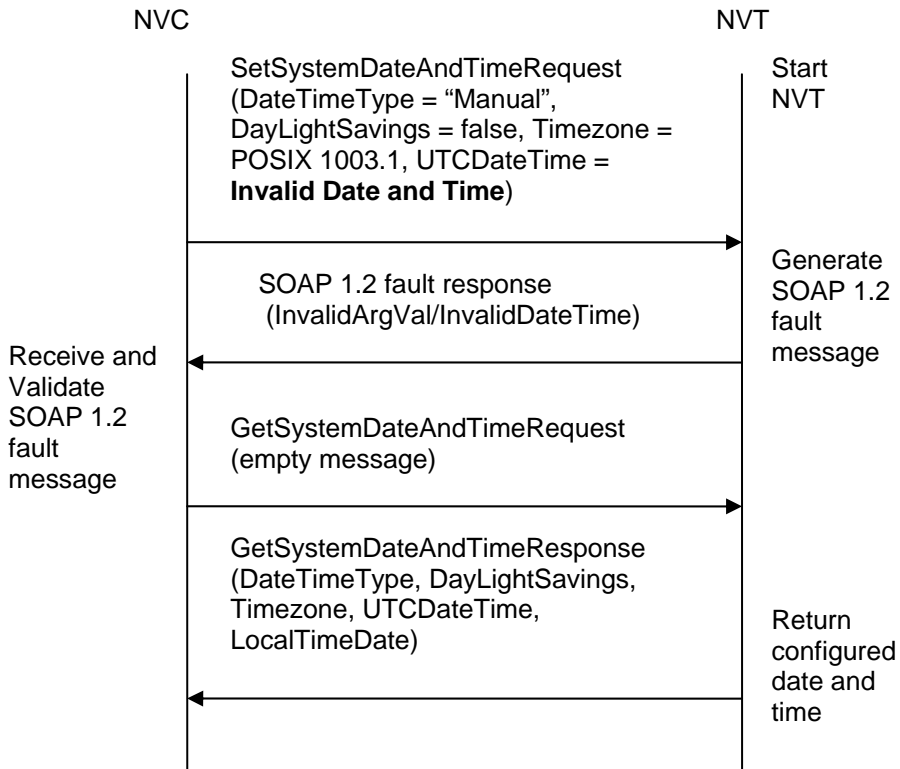
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify the behaviour of NVT for invalid system date and time configuration.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetSystemDateAndTimeRequest message with invalid Date and Time (DateTimeType = "Manual", DayLightSavings = false, Timezone = POSIX 1003.1, UTCDateTime = **Invalid Date and Time**).

4. Verify that NVT generates SOAP 1.2 fault response (InvalidArgVal/InvalidDateTime).

5. Verify the NVT system date and time configurations through GetSystemDateAndTimeRequest message.

4. NVT sends system date and time configurations in the GetSystemDateAndTimeResponse message (DateTimeType = Manual or NTP, DayLightSavings = true or false, Timezone = POSIX 1003.1, UTC DateTime = Hour:Min:Sec, Year:Month:Day and LocalTimeDate = Hour:Min:Sec, Year:Month:Day).

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct fault code in the SOAP fault message (InvalidArgVal/ InvalidDateTime).

The DUT did not GetSystemDateAndTimeResponse message.

The DUT returned “Invalid Date and Time” in the GetSystemDateAndTimeResponse message.

**Note:** If system date and time are set manually, then DUT MUST return UTCDateTime or LocalDateTime in the GetSystemDateAndTimeResponse message.

See Annex A for Invalid SOAP 1.2 fault message definition.

### 8.2.12 NVT SYSTEM COMMAND FACTORY DEFAULT

**Test Label:** Device Management NVT System Command **SetSystemFactoryDefault** Test.

**Command Under Test:** SetSystemFactoryDefault

**ONVIF Core Specification Coverage:** 8.3.6 Factory default

**Device Type:** NVT

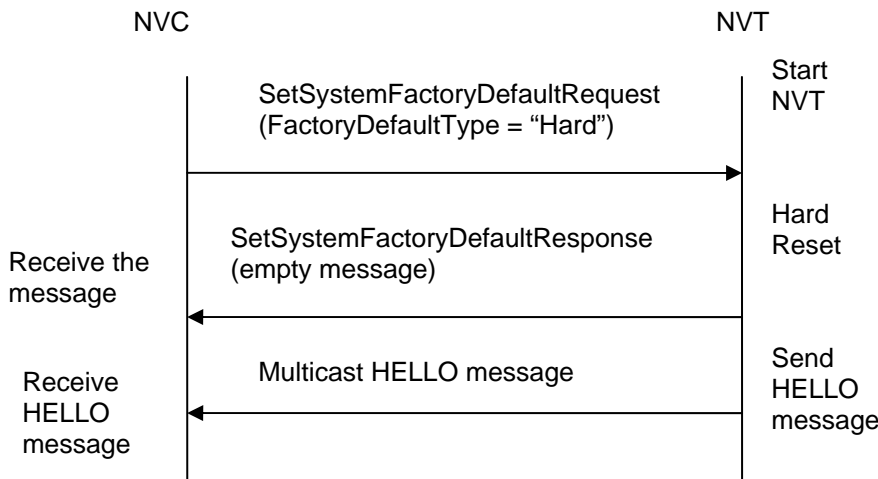
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To reload all parameters of NVT to their default values through **SetSystemFactoryDefault** command. This test is for hard factory default.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetSystemFactoryDefaultRequest message (**FactoryDefaultType = “Hard”**).

4. Verify that NVT sends SetSystemFactoryDefaultResponse message.

5. Verify that NVT sends Multicast HELLO message after hard reset.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SetSystemFactoryDefaultResponse message.

The DUT did not send HELLO message.

**Note:** After Hard Reset certain DUT's are not IP reachable. In such situation, DUT must be configured with an IPv4 address, must be IP reachable in the test network and other relevant configurations to be done for further tests.

### 8.2.12.1 NVT SYSTEM COMMAND FACTORY DEFAULT

**Test Label:** Device Management NVT System Command **SetSystemFactoryDefault** Test.

**Command Under Test:** SetSystemFactoryDefault

**ONVIF Core Specification Coverage:** 8.3.6 Factory default

**Device Type:** NVT

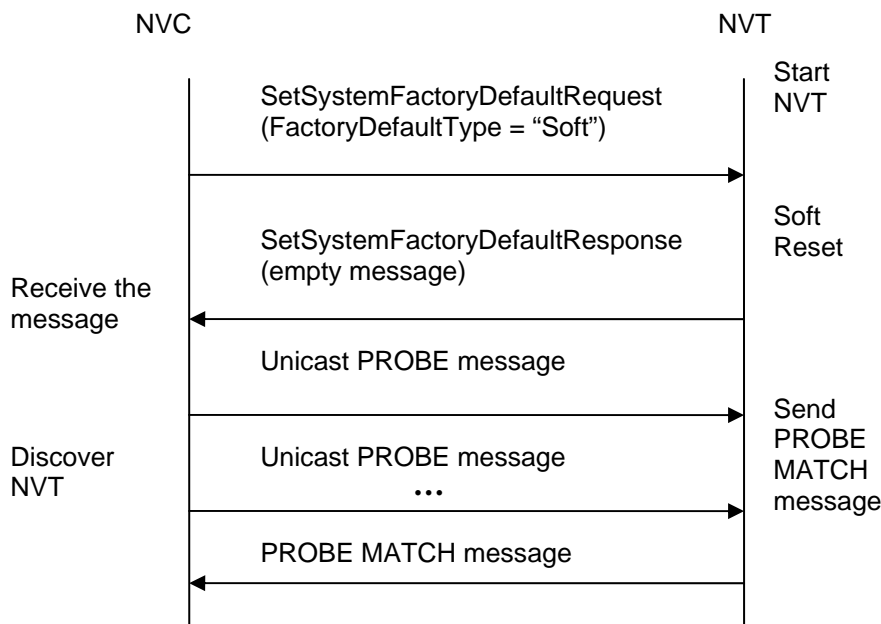
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To reload all parameters of NVT to their factory default values through **SetSystemFactoryDefault** command. This test is for soft factory default.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SetSystemFactoryDefaultRequest message (**FactoryDefaultType = "Soft"**).

4. Verify that NVT sends SetSystemFactoryDefaultResponse message.

5. NVC will verify that NVT is accessible after soft reset. NVC will send Unicast PROBE message several times (i.e. 50 times at an interval of 5 seconds).

6. Verify that NVT sends a PROBE MATCH message.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SetSystemFactoryDefaultResponse message.

The DUT did not send PROBE MATCH message (i.e. DUT cannot be discovered).

### 8.2.13 NVT SYSTEM COMMAND RESET

**Test Label:** Device Management NVT System Command **SystemReboot** Test.

**Command Under Test:** SystemReboot

**ONVIF Core Specification Coverage:** 8.3.10 Reboot

**Device Type:** NVT

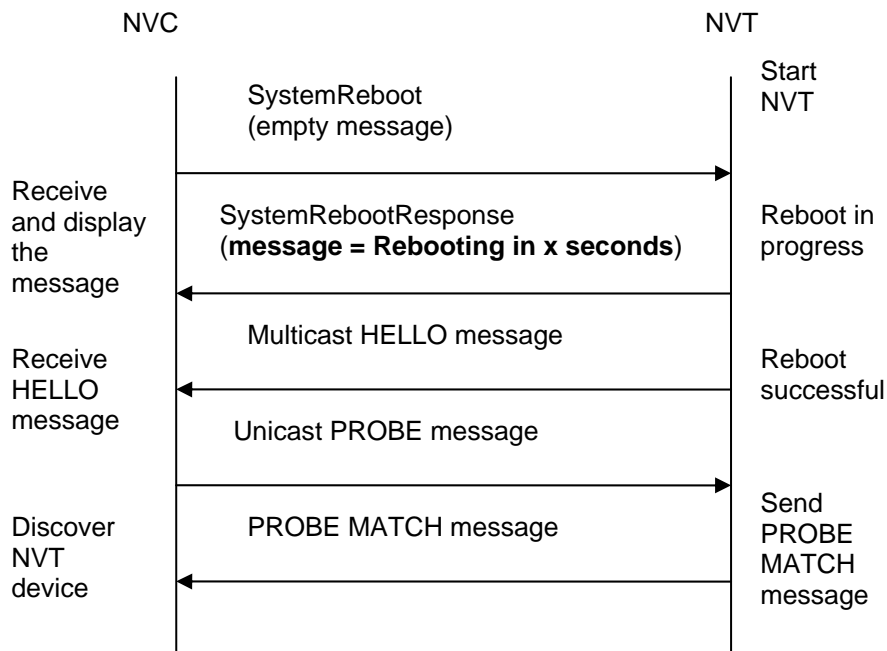
**Requirement Level:** MUST

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To reboot the NVT through **SystemReboot** command.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke SystemReboot message to reset the NVT.

4. Verify that NVT sends SystemRebootResponse message (**example message string = “Rebooting in x seconds”**).

5. NVT will send Multicast HELLO message after it is successfully rebooted.

6. NVC will verify the HELLO message sent by NVT.

7. NVC will send Unicast PROBE message to discover the NVT.

8. NVT will send a PROBE MATCH message.

9. NVC will verify the PROBE MATCH message sent by NVT.

**Note:** if BYE message is supported by NVT, then NVT shall send multicast BYE message before the reboot.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send SystemRebootResponse message.

The DUT did not send HELLO message.

The DUT did not send PROBE MATCH message between 0 to 500 milliseconds.

The DUT did not send PROBE MATCH message.

### 8.3 Media Configuration Test Cases

This section covers tests designed for NVT Media Configuration Feature.

#### 8.3.1 NVT MEDIA PROFILE CONFIGURATION

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

**Command Under Test:** GetProfiles

**ONVIF Core Specification Coverage:** 10.2.2 Get media profiles

**Device Type:** NVT

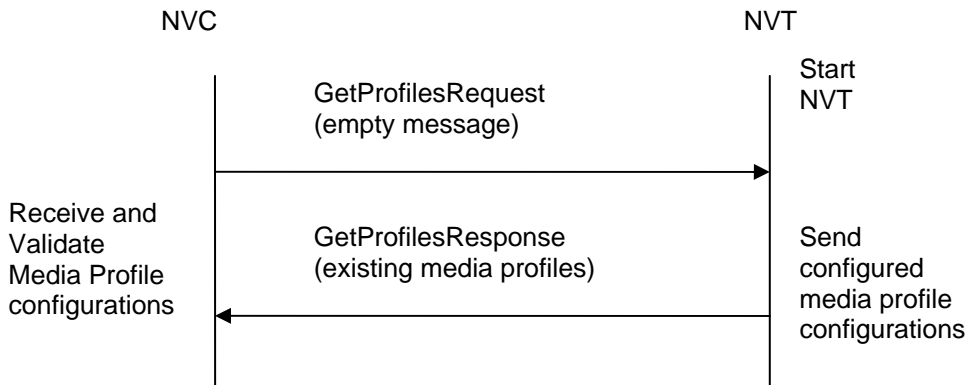
**Requirement Level:** MUST

**WSDL Reference:** media.wSDL

**Test Purpose:** To retrieve existing media profile configurations of NVT and the corresponding media entities (video source and video encoder).

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetProfilesRequest message to retrieve existing media profiles configurations of the NVT.

4. Verify that the NVT returns at-least one media profile with video configuration (video source and video encoder) in the GetProfilesResponse message.

**Test Result: PASS** – DUT passes all assertions.

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

The DUT has no default media profile configuration.

The DUT did not support Video Source Configuration in one or more media profiles.

The DUT did not support Video Encoder Configuration in one or more media profiles.

### 8.3.2 NVT DYNAMIC MEDIA PROFILE CONFIGURATION

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

**ONVIF Core Specification Coverage:** 10.2.1 Create media profile, 10.2.2 Get media profiles , 10.2.3 Get media profile, 10.2.4 Add video source configuration to a profile, 10.2.5 Add video encoder configuration to a profile, 10.2.11 Remove video source configuration from a profile, 10.2.12 Remove video encoder configuration from a profile, 10.2.18 Delete media profile

**Device Type:** NVT

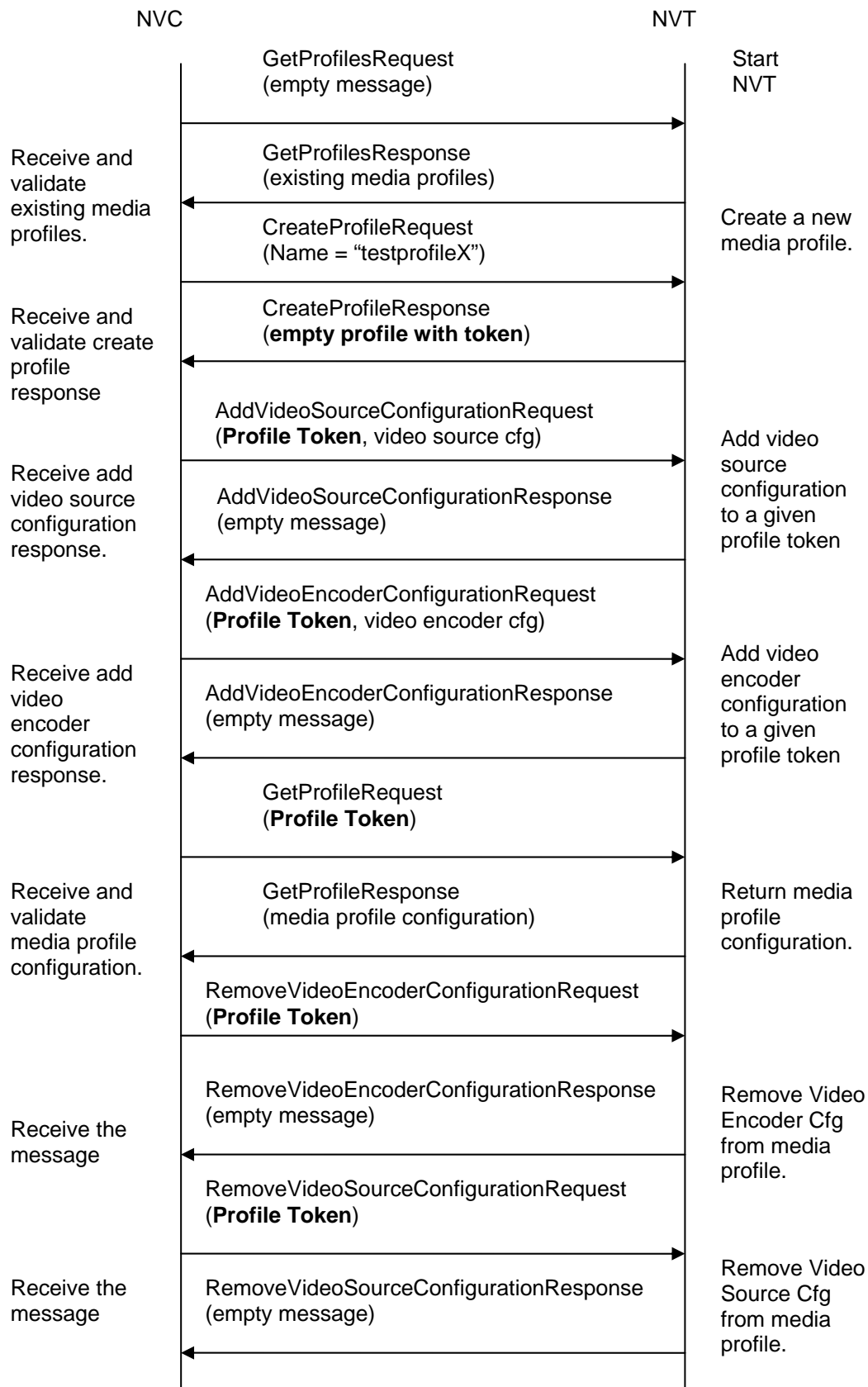
**Requirement Level:** MUST

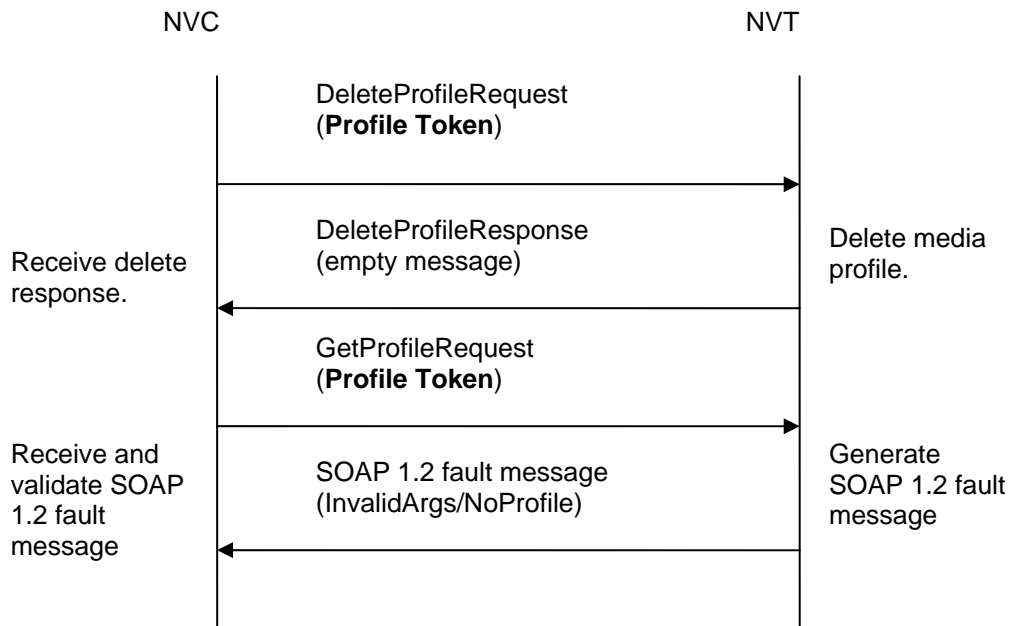
**WSDL Reference:** media.wsdl

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

**Test Configuration:** NVC and NVT

**Test Sequence:**





**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetProfilesRequest message to retrieve existing media profiles configurations of the NVT.

4. Verify that the NVT returns at-least one media profile with video configuration (video source and video encoder) in GetProfilesResponse message.

5. NVC will invoke CreateProfileRequest message to create a new empty media profile.

6. NVT returns an empty profile with no profile entities in the CreateProfileResponse message.

7. NVC will invoke AddVideoSourceConfigurationRequest message (**Profile Token**, Reference to **Video Source Configuration of existing media profile**) to add video source configuration to new profile.

8. NVT sends AddVideoSourceConfigurationResponse message indicating successfully addition of video source configuration.

9. NVC will invoke AddVideoEncoderConfigurationRequest message (**Profile Token**, Reference to **Video Encoder Configuration of existing media profile**) to add video encoder configuration to new profile.

10. NVT sends AddVideoEncoderConfigurationResponse message indicating successfully addition of video encoder configuration.

11. NVC will invoke GetProfileRequest (**Profile Token**) message to verify video source and encoder configurations in a new profile.

12. NVT will return media profile configuration for requested media profile in the GetProfileResponse message.

13. NVC will invoke RemoveVideoEncoderConfigurationRequest message (**Profile Token**) to remove video encoder configuration from a media profile.

14. NVT sends RemoveVideoEncoderConfigurationResponse message indicating successfully removal of video encoder configuration.

15. NVC will invoke RemoveVideoSourceConfigurationRequest message (**Profile Token**) to remove video source configuration from a media profile.

16. NVT sends RemoveVideoSourceConfigurationResponse message indicating successfully removal of video source configuration.

17. NVC will invoke DeleteProfileRequest (**Profile Token**) message to delete the newly created media profile.

18. NVT will delete the media profile and sends DeleteProfileResponse message.

19. NVC will invoke GetProfileRequest (**deleted Profile Token**) message to check the existence of deleted media profile.

20. NVT will generate SOAP 1.2 fault message (InvalidArgs/NoProfile).

**Test Result: PASS** – DUT passes all assertions.

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

The DUT did not send CreateProfileResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send GetProfileResponse message.

The DUT did not send RemoveVideoEncoderConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

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

**Note:** See Annex A for Invalid SOAP 1.2 fault message definition.

### 8.3.3 NVT JPEG VIDEO ENCODER CONFIGURATION

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

**ONVIF Core Specification Coverage:** 10.2.2 Get media profiles, 10.5.2 Get video encoder configuration, 10.5.5 Modify a video encoder configuration.

**Device Type:** NVT

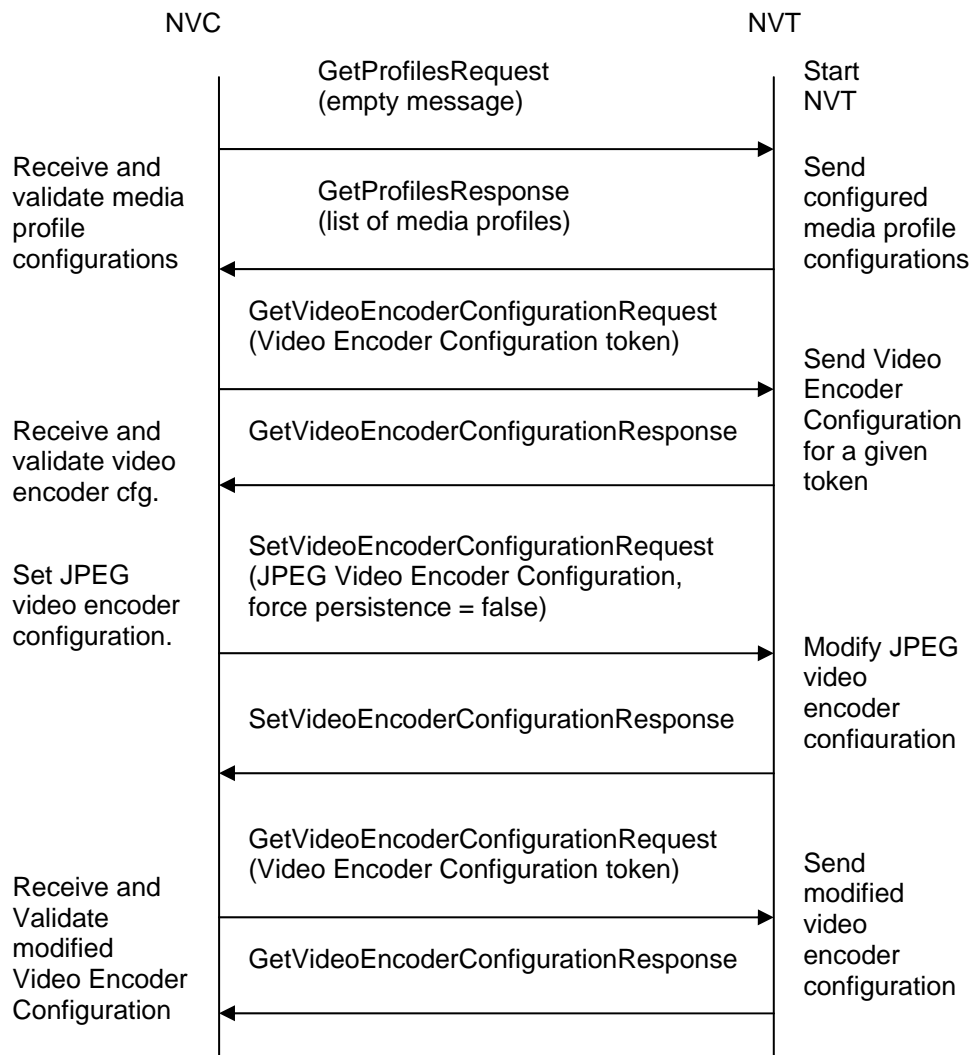
**Requirement Level:** MUST

**WSDL Reference:** media.wsdl

**Test Purpose:** To verify NVT JPEG Video Encoder Configurations Setting.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetProfilesRequest message to retrieve existing media profiles configurations of the NVT.

4. Verify that the NVT returns at-least one media profile with video configuration (video source and video encoder) in GetProfilesResponse message.

5. NVC will invoke GetVideoEncoderConfigurationRequest message (**Video Encoder Configuration Token of existing media profile**) to retrieve video encoder configuration for a given video encoder configuration token.

6. NVT sends requested Video Encoder Configuration in the GetVideoConfigurationResponse message.

7. NVC will invoke SetVideoEncoderConfiguration message (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**).

8. NVT modifies JPEG video encoder configuration and responds with SetVideoConfigurationResponse message indicating success.

9. NVC will verify the JPEG Video Encoder Configurations settings on NVT by GetVideoEncoderConfigurationRequest message.

9. NVT sends modified JPEG Video Encoder Configurations in the GetVideoConfigurationResponse message (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**).

**Test Result: PASS** – DUT passes all assertions.

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

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send SetVideoEncoderConfigurationResponse message.

The DUT did not modify JPEG Video Encoder Configurations.

#### 8.3.4 NVT MEDIA STREAM URI – RTP/UDP UNICAST TRANSPORT

**Test Label:** Media Configuration NVT Media Stream URI Request.

**Command Under Test:** GetStreamUri

**ONVIF Core Specification Coverage:** 10.11.1 Request stream URI

**Device Type:** NVT

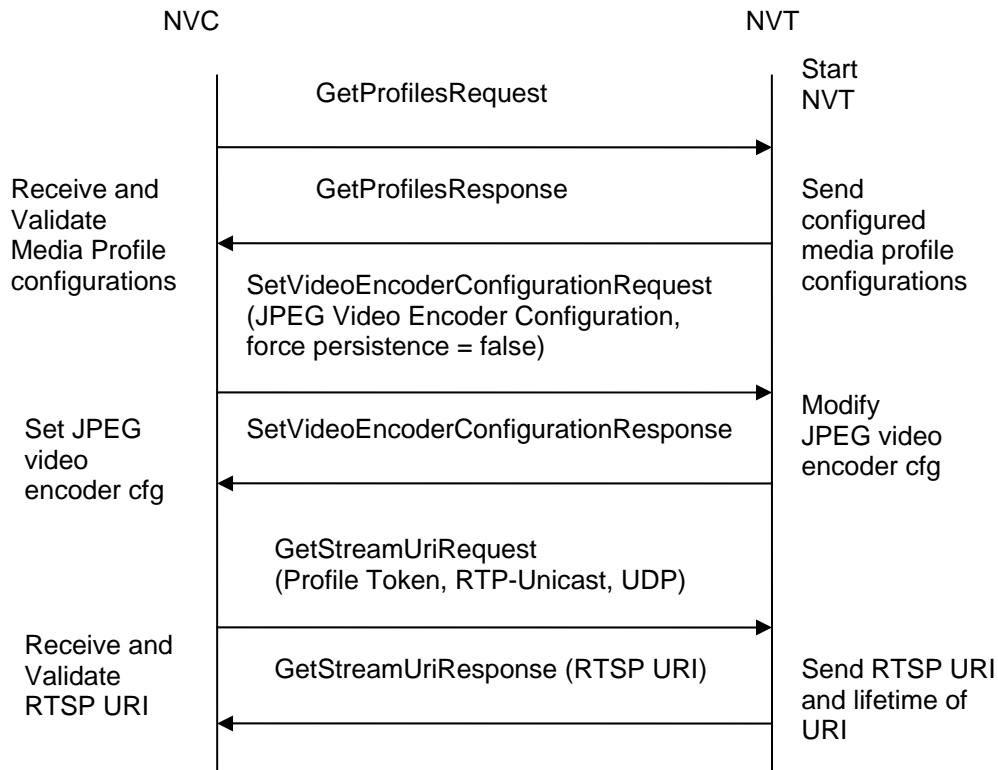
**Requirement Level:** MUST

**WSDL Reference:** media.wsdl

**Test Purpose:** To retrieve media control stream URI of NVT for a given media profile.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetProfilesRequest message to retrieve existing media profile configurations.

4. NVT returns existing media profile configurations in the GetProfilesResponse message.

5. NVC will invoke SetVideoEncoderConfiguration message (**Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**).

6. NVT modifies JPEG video encoder configuration and responds with SetVideoConfigurationResponse message indicating success.

7. NVC will invoke GetStreamUriRequest message (**Profile Token, RTP-Unicast, UDP transport**) to retrieve media stream URI for a given media profile.

8. NVT sends **RTSP URI** and parameters defining the lifetime of the URI like **ValidUntilConnect, ValidUntilReboot and Timeout** in the GetStreamUriResponse message.

9. NVC verifies the RTSP media stream URI provided by the NVT.

**Test Result: PASS** – DUT passes all assertions

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

The DUT did not support Video Encoder Configuration in one or more media profiles.

The DUT did not send SetVideoConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

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

**Note:** See Annex A for usage of ValidUntilConnect, ValidUntilReboot, and Timeout parameters.

See Annex A for correct syntax for the StreamSetup element in GetStreamUri requests.

### 8.3.5 NVT MEDIA STREAM URI – RTP/RTSP/HTTP TRANSPORT

**Test Label:** Media Configuration NVT Media Stream URI Request.

**Command Under Test:** GetStreamUri

**ONVIF Core Specification Coverage:** 10.11.1 Request stream URI

**Device Type:** NVT

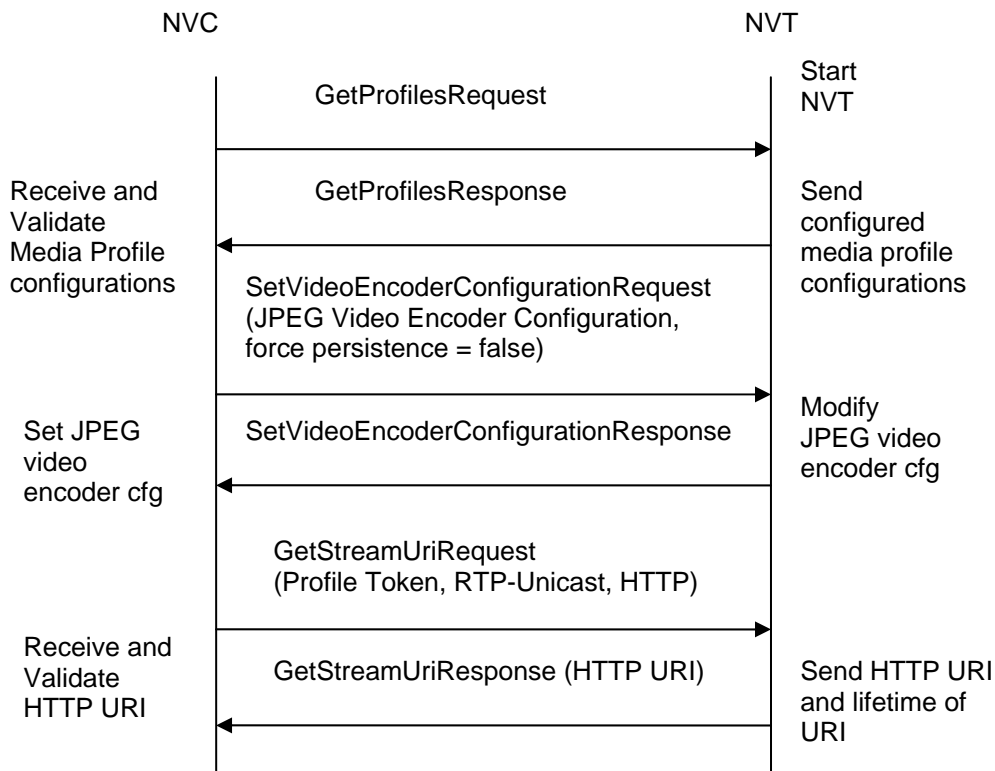
**Requirement Level:** MUST

**WSDL Reference:** media.wsdl

**Test Purpose:** To retrieve media control stream URI of NVT for a given media profile.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetProfilesRequest message to retrieve existing media profile configurations.

4. NVT returns existing media profile configurations in the GetProfilesResponse message.

5. NVC will invoke SetVideoEncoderConfiguration message ( **Encoding = "JPEG", Resolution = ["Width", "Height"], Quality = q1, Session Timeout = t1 and force persistence = false**).

6. NVT modifies JPEG video encoder configuration and responds with SetVideoConfigurationResponse message indicating success.

7. NVC will invoke GetStreamUriRequest message (**Profile Token, RTP-Unicast, HTTP transport**) to retrieve media stream URI for a given media profile.

8. NVT sends **HTTP URI** and parameters defining the lifetime of the URI like **ValidUntilConnect, ValidUntilReboot** and **Timeout** in the GetStreamUriResponse message.

9. NVC verifies the HTTP media stream URI provided by the NVT.

**Test Result: PASS** – DUT passes all assertions

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

The DUT did not support Video Encoder Configuration in one or more media profiles.

The DUT did not send SetVideoConfigurationResponse message.

The DUT did not send GetStreamUriResponse message.

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

**Note:** See Annex A for usage of ValidUntilConnect, ValidUntilReboot, and Timeout parameters.

See Annex A for correct syntax for the StreamSetup element in GetStreamUri requests.

### 8.3.6 NVT SOAP FAULT MESSAGE

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

**Command Under Test:** GetStreamUri

**ONVIF Core Specification Coverage:** 10.11.1 Request stream URI

**Device Type:** NVT

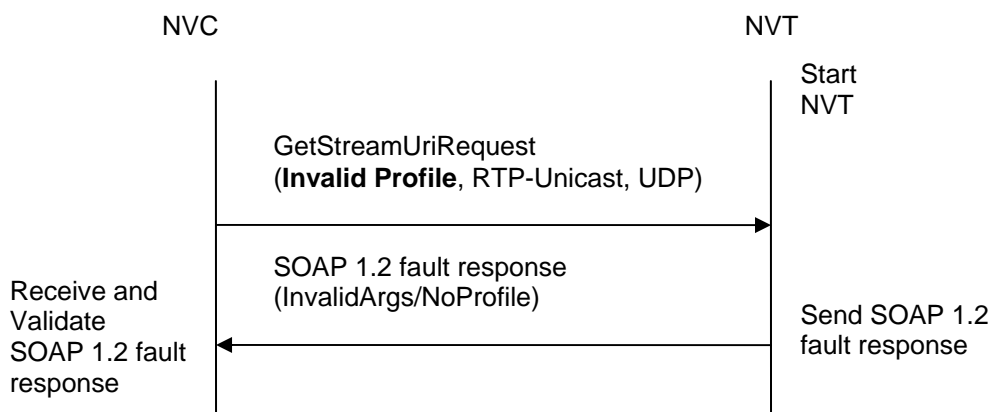
**Requirement Level:** MUST

**WSDL Reference:** media.wsdl

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

**Test Configuration:** NVC and NVT.

**Test Sequence:**



**Test Procedure:** 1. Start an NVC.

2. Start an NVT.

3. NVC will invoke GetStreamUriRequest message with **invalid media profile**.

4. NVT will generate the SOAP 1.2 fault message (InvalidArgs/NoProfile).

**Test Result: PASS** – DUT passes all assertions

**FAIL** – The DUT did not send SOAP 1.2 fault message.

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

**Note:** See Annex A for Invalid Media Profile and SOAP 1.2 fault message definitions.

See Annex A for correct syntax for the StreamSetup element in GetStreamUri requests.

#### 8.3.6.1 NVT SOAP FAULT MESSAGE

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

**Command Under Test:** GetStreamUri

**ONVIF Core Specification Coverage:** 10.11.1 Request stream URI

**Device Type:** NVT

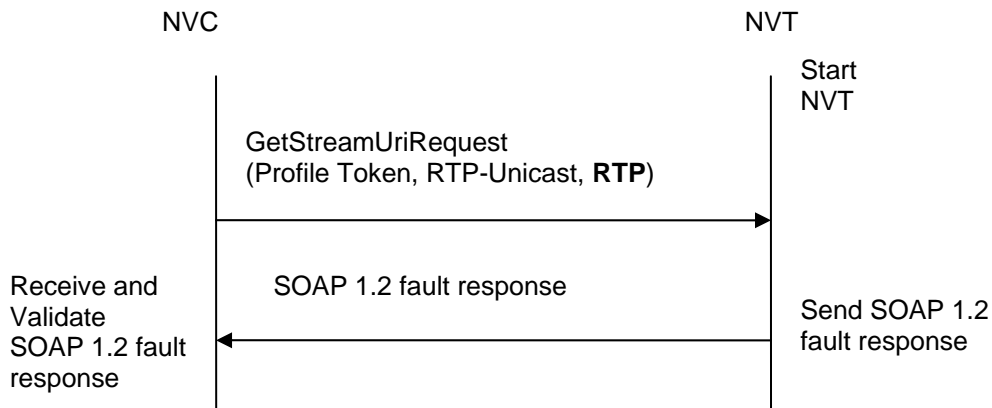
**Requirement Level:** MUST

**WSDL Reference:** media.wsdl

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

**Test Configuration:** NVC and NVT.

**Test Sequence:**



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

**Test Result: PASS** – DUT passes all assertions

**FAIL** – The DUT did not send SOAP 1.2 fault message.

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

**Note:** See Annex A for Invalid SOAP 1.2 fault message definition.

See Annex A for correct syntax for the StreamSetup element in GetStreamUri requests.

## 8.4 Real Time Viewing Test Cases

This section covers tests designed for NVT Real Time Viewing Feature. All Real Time Viewing test cases require RTSP Control Stream URI which is retrieved by **GetStreamUriRequest** message.

### 8.4.1 NVT MEDIA CONTROL – RTSP/TCP

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

**ONVIF Core Specification Coverage:** 11.2.1 Stream control, 11.2.1.1 RTSP

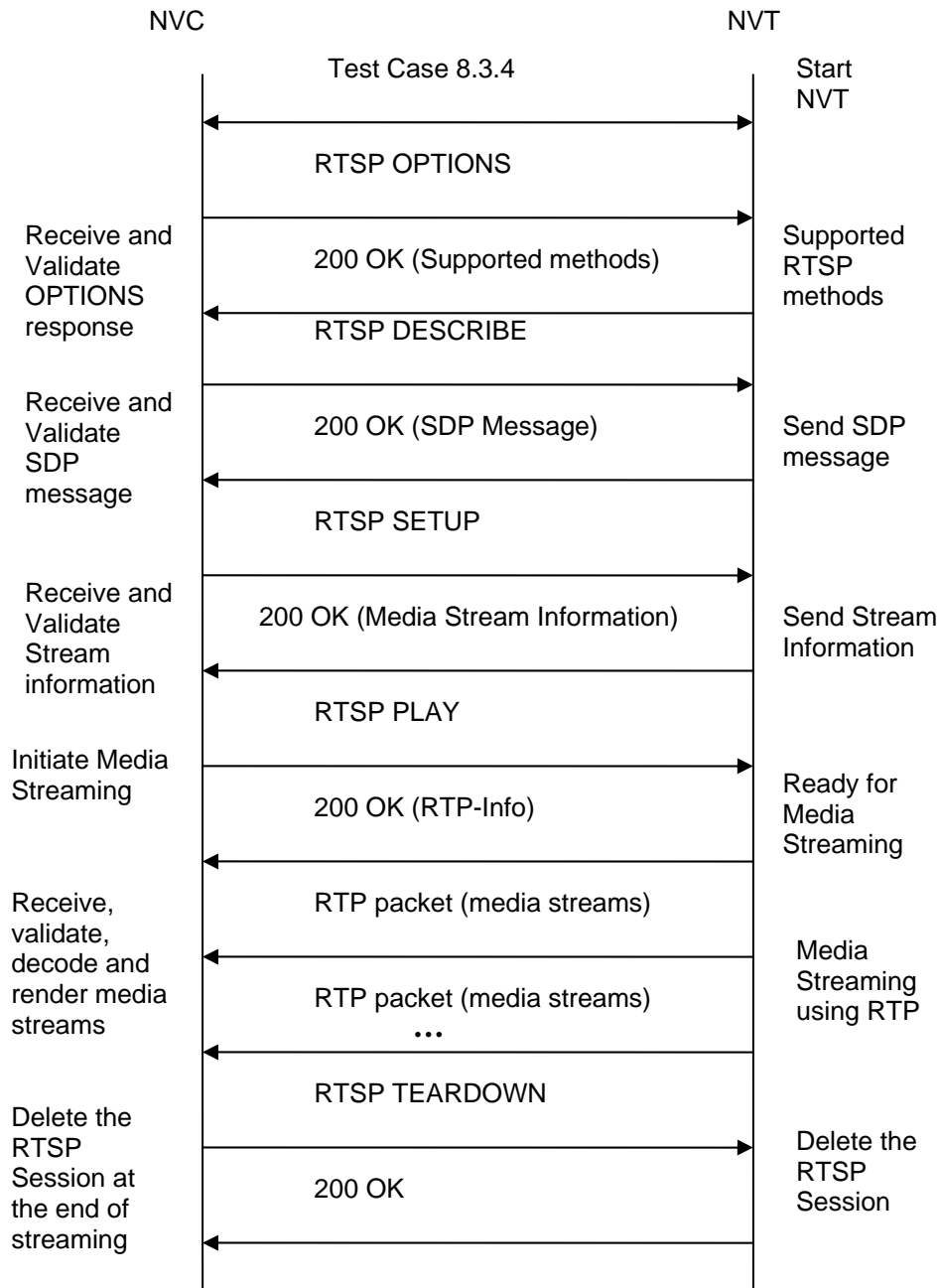
**Device Type:** NVT

**Requirement Level:** MUST

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

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Execute test case 8.3.4 and retrieve RTSP URI.

2. NVC will invoke RTSP OPTIONS control request to understand the RTSP methods supported by NVT.

3. NVT sends 200 OK Response and list of supported RTSP methods.

4. NVC will invoke RTSP DESCRIBE control request to retrieve the media description information.

5. NVT sends 200 OK Response and SDP message.

6. NVC validates the session description information in the SDP message.
7. NVC will invoke RTSP SETUP control request to create a RTSP Session.
8. NVT sends 200 OK Response and Stream Information details.
9. NVC Verifies “**Transport**”, “**Session**” and “**Timeout**” header fields in the SETUP response message.
10. NVC will invoke RTSP PLAY control request to initiate the media streaming.
11. NVT sends 200 OK Response and RTP protocol information.
12. NVC verifies “**Session**”, “**RTP-Info**”, “**seq**” and “**rtptime**” header fields in the PLAY response message.
13. NVT transfers media streams over RTP/UDP.
14. NVC validates RTP header for each media stream and render it after the validation.
15. NVC will invoke RTSP TEARDOWN control request to terminate the RTSP session at the end of the streaming.
16. NVT sends 200 OK Response and terminates the RTSP Session.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send correct media stream information in the SDP message.

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

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

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

RTSP Session is terminated by DUT during media streaming.

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

**Note:** See Annex A for Invalid RTP header definition.

#### 8.4.2 NVT MEDIA STREAMING – RTP/UDP UNICAST TRANSPORT

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

**ONVIF Core Specification Coverage:** 11.1.1.1 RTP/UDP, 11.1.2.1 RTP

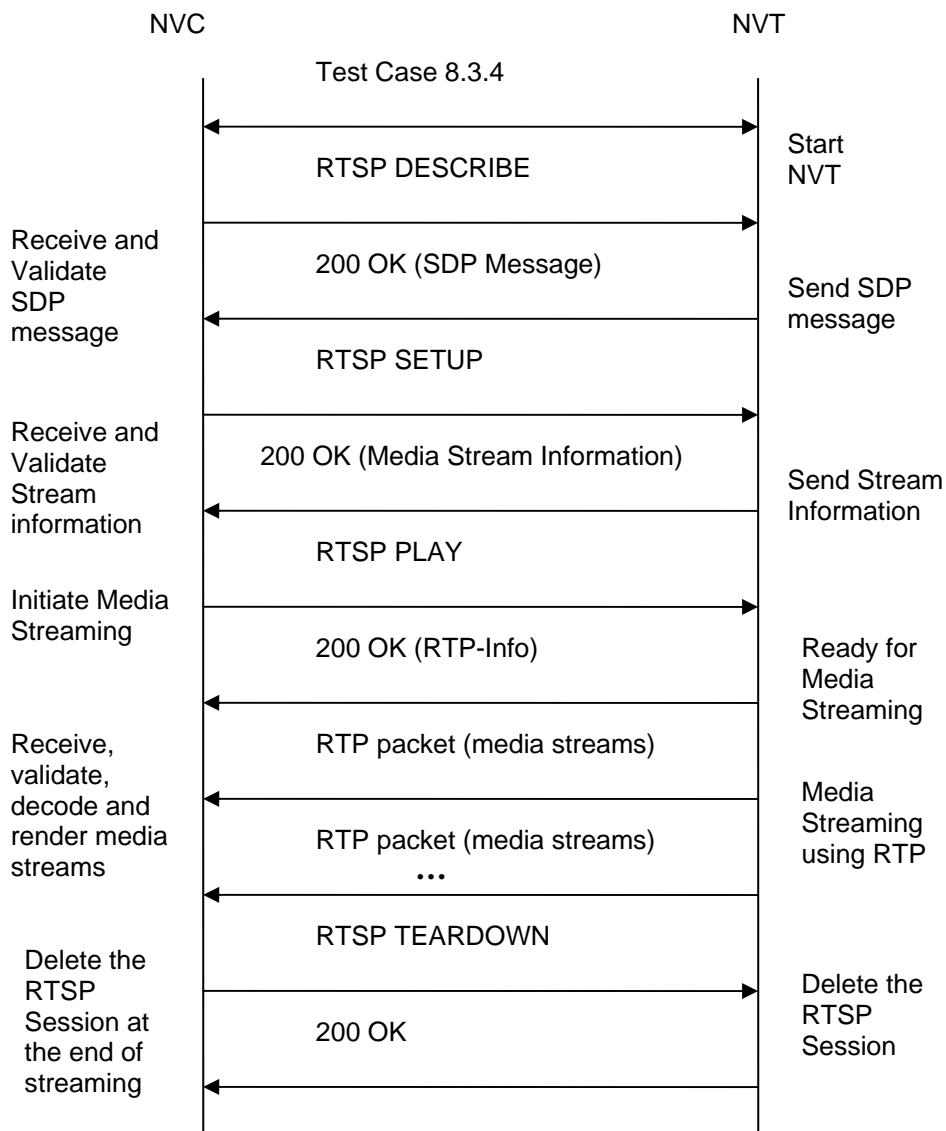
**Device Type:** NVT

**Requirement Level:** MUST

**Test Purpose:** To verify NVT media streaming based on RTP/UDP Unicast transport.

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Execute test case 8.3.4 and retrieve RTSP URI.

2. NVC will invoke RTSP control requests (DESCRIBE, SETUP and PLAY).

3. NVT sends 200 OK Response to RTSP controls requests.

4. NVC will receive RTP/UDP streams from NVT and validate the RTP header for each media stream.

5. NVC will decode the media stream and render it.

6. NVC will invoke RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

7. NVT sends 200 OK Response and terminates the RTSP Session.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – RTSP Session is terminated by DUT during media streaming.

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

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

**Note:** See Annex A for Invalid RTP header definition.

#### 8.4.3 NVT MEDIA STREAMING – RTP/RTSP/HTTP TRANSPORT

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

**ONVIF Core Specification Coverage:** 11.1.1.4 RTP/RTSP/HTTP/TCP, 11.2.1.2 RTSP over HTTP

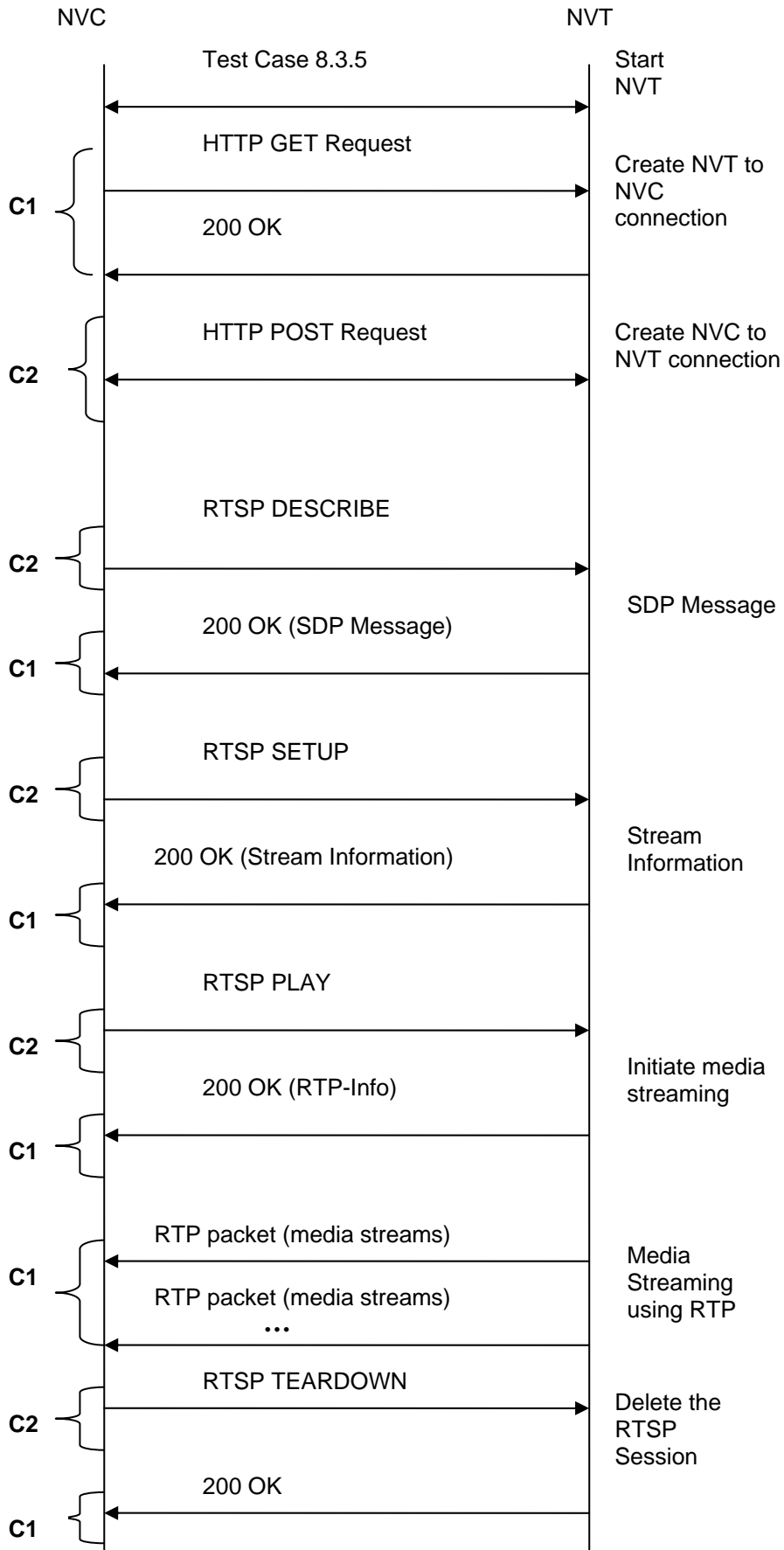
**Device Type:** NVT

**Requirement Level:** MUST

**Test Purpose:** To verify NVT media streaming based on HTTP transport.

**Test Configuration:** NVC <->HTTP Proxy <-> NVT.

**Test Sequence:**



**Test Procedure:** 1. Execute test case 8.3.5 and retrieve HTTP URI.

2. NVC will invoke HTTP GET Request on NVT and establishes NVT to NVC connection for RTP data transfer.

3. NVC will invoke HTTP POST Request and establishes NVC to NVT connection for RTSP control requests.

4. NVC will invoke RTSP control requests (DESCRIBE, SETUP and PLAY) on HTTP POST connection.

5. NVT sends RTSP 200 OK response on HTTP GET connection.

6. NVT transfers RTP media streams on HTTP GET connection.

7. NVC receives RTP media streams on HTTP GET connection and validates the RTP header.

8. NVC will decode the media stream and render it.

9. NVC will invoke RTSP TEARDOWN control request on HTTP POST connection and closes HTTP POST connection.

10. NVC will close HTTP GET connection at the end of the media streaming.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – RTSP Session is terminated by DUT during media streaming.

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

HTTP Session is terminated by DUT during media streaming.

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

**Note:** See Annex A for Invalid RTP header definition.

#### 8.4.4 NVT MEDIA STREAMING – RTSP KEEPALIVE

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

**ONVIF Core Specification Coverage:** 11.2.1.1.1 Keep-alive method for RTSP session

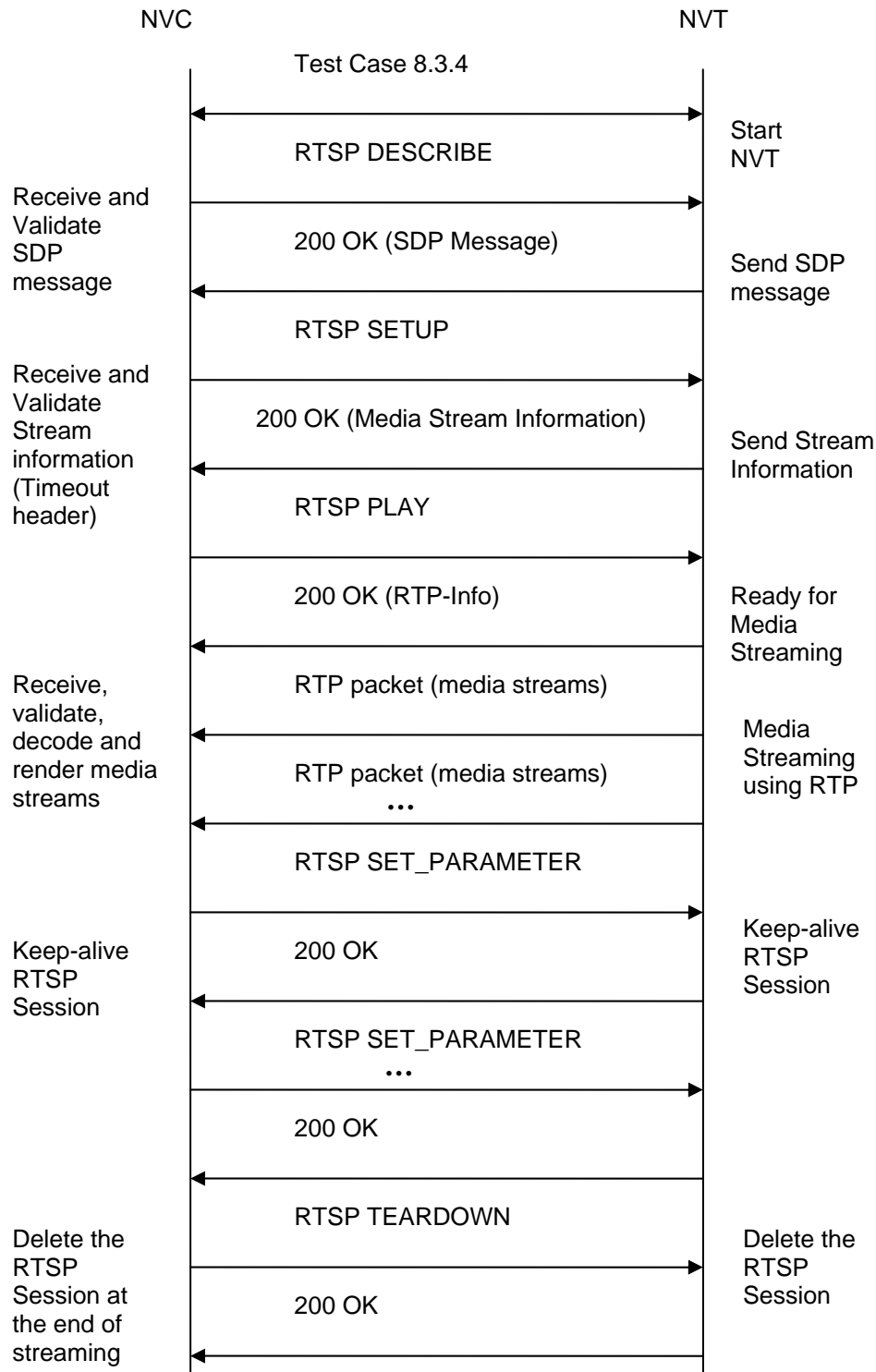
**Device Type:** NVT

**Requirement Level:** MUST

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

**Test Configuration:** NVC and NVT

**Test Sequence:**



**Test Procedure:** 1. Execute test case 8.3.4 and retrieve RTSP URI.

2. NVC will invoke RTSP control requests (DESCRIBE, SETUP and PLAY).

3. NVC will verify “**Timeout**” header in the SETUP Response from NVT.

4. Based on the “**Timeout**” value, NVC will invoke RTSP SET\_PARAMETER messages.
5. NVT will respond with 200 OK for RTSP SET\_PARAMETER request.
6. Verify that the NVC and NVT are exchanging periodic SET\_PARAMETER messages while a stream is being delivered.

**Test Result: PASS** – DUT passes all assertions.

**FAIL** – The DUT did not send Timeout header in RTSP SETUP RESPONSE.

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

The DUT terminates the RTSP Session during media streaming.

## Annex A (informative)

This section describes the meaning of the following definitions, these definitions are used in the description of the test procedure (Section 8.0).

### A.1 Invalid Device Type and Scope Type

Device Type in the <d:Types:> declaration: dn:NetworkVideoTransmitter

Anything other than “**NetworkVideoTransmitter**” is considered as Invalid Device Type.

Invalid Scope Type:

- Scope URI is not formed according to the rules of RFC 3986.

### A.2 Invalid Hostname, DNSname

A string which is not formed according to the rules of RFC 952 is considered as invalid string.

### A.3 Invalid Media Profile

Media profile token is a string of max length value of 64.

Invalid Media Profile:

- A string which is not formed according to the rules of RFC 952.
- A string which exceeds max length value of 64.

### A.4 Invalid TimeZone

The Time Zone format is specified by POSIX, refer to POSIX 1003.1 section 8.3.

Example: Europe, Paris TZ=CET-1CEST,M3.5.0/2,M10.5.0/3

CET = designation for standard time when daylight saving is not in force.

-1 = offset in hours = negative so 1 hour east of Greenwich meridian.

CEST = designation when daylight saving is in force ("Central European Summer Time")

, = no offset number between code and comma, so default to one hour ahead for daylight saving

M3.5.0 = when daylight saving starts = the last Sunday in March (the "5th" week means the last in the month)

/2, = the local time when the switch occurs = 2 a.m. in this case

M10.5.0 = when daylight saving ends = the last Sunday in October.

/3, = the local time when the switch occurs = 3 a.m. in this case

A TimeZone token which is not formed according to the rules of POSIX 1003.1 section 8.3 is considered as invalid timezone.

### A.5 Invalid RTP Header

A RTP header which is not formed according to the header field format defined in the RFC 3550 Section 5.1 is considered as invalid RTP header.

## A.6 Invalid SOAP 1.2 Fault Message

A SOAP 1.2 fault message which is not formed according to the rules defined in SOAP 1.2, Part 1 Section 5.4 is considered as invalid.

A SOAP 1.2 fault message which does not include ONVIF defined namespace “**ter=http://www.onvif.org/ver10/error**” is considered as invalid.

## A.7 Usage of URI Life Time

GetStreamUriResponse message contains the Uri to be used for requesting the media stream as well as parameters defining the lifetime of the Uri.

Valid combinations (definition of the lifetime of the Uri):

- ValidUntilConnect = true, ValidUntilReboot = false, Timeout is zero
- ValidUntilConnect = true, ValidUntilReboot = false, Timeout is non-zero
- ValidUntilConnect = false, ValidUntilReboot = true, Timeout is zero
- ValidUntilConnect = false, ValidUntilReboot = false, Timeout is non-zero

GetStreamUriResponse message which does not include any of the above valid combination is considered as invalid.

## A.8 Invalid WSDL URL

An URL which is not formed according to the rules of RFC 3986 is considered as invalid WSDL URL.

## A.9 Valid/Invalid IPv4 Address

IPv4 Address token is represented in dotted decimal notation (32 bit internet address is divided into four 8 bit fields and each field is represented in decimal number separated by a dot).

- Valid IPv4 addresses are in the range 0.0.0.0 to 255.255.255.255 excluding 0/8, 255/8, and 127/8, as defined in RFC 758, and 169.254/16 as defined in RFC 3927.
- Valid IPv4 addresses for a device must be valid according to the defined network mask and gateway (the gateway must be reachable and must not be identical to the assigned IPv4 address).
- Reserved addresses such as 240.0.0.0 through 255.255.255.254, as defined in RFC 2780 are prohibited for IPv4 devices.

## A.10 StreamSetup syntax for GetStreamUri

The following media stream setups for GetStreamUri are covered in this Test Specification:

1. RTP unicast over UDP
2. RTP over RTSP over HTTP over TCP

The correct syntax for the StreamSetup element for these media stream setups are as follows:

1. RTP unicast over UDP

```
<StreamSetup>  
  <StreamType>RTP_unicast</StreamType>  
  <Transport>  
    <Protocol>UDP</Protocol>  
  </Transport>  
</StreamSetup>
```

2. RTP over RTSP over HTTP over TCP

```
<StreamSetup>  
  <StreamType>RTP_unicast</StreamType>  
  <Transport>  
    <Protocol>HTTP</Protocol>  
  </Transport>  
</StreamSetup>
```