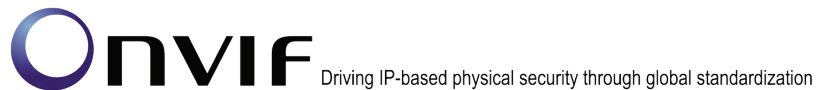


# ONVIF™

## Profile Q Test Specification

Version 16.01

January 2016



© 2016 by ONVIF, Inc. All rights reserved.

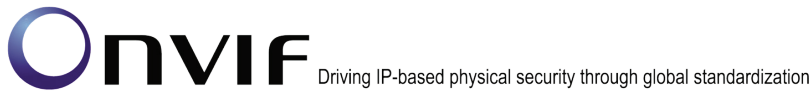
Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

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

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

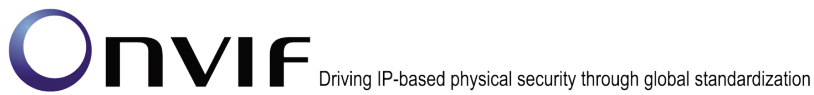
### Revision History

Ver.	Date	Description
14.12		First issue
15.06	June, 2015	<p>The following test cases were updated:</p> <ul style="list-style-type: none"> <li>Factory Default state verification</li> <li>Default access policy - Operator</li> <li>Default access policy - Administrator and Anonymous</li> <li>Default access policy - Administrator and User/Operator</li> <li>Processor Usage event</li> <li>Last Reboot event (Status Change)</li> <li>Firmware Upload</li> </ul>
16.01	January, 2016	<p>The parameter value "UTCDateTime" has been changed in step 7 of section 4.3.7 and step 10 of section 4.3.6</p> <p>Clarifications have been added to steps 6 and 7 of section 4.3.4. The FAIL condition:"The DUT did not send UnsubscribeResponse message." has been removed from test QUICK_INSTALL-3-1-4 section 4.3.4</p> <p>The steps 32-34 have been added to section 4.1.1 (QUICK_INSTALL-1-1-1)</p> <p>The steps 8-10 have been added to section 4.4.2 (QUICK_INSTALL-4-1-2)</p>

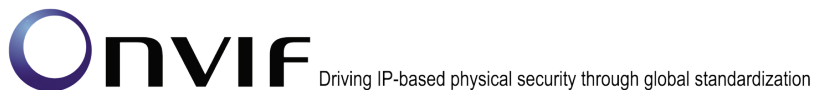


## Table of Contents

1	Introduction .....	6
1.1	Scope .....	6
1.1.1	General .....	6
1.1.2	Default Access Policy .....	7
1.1.3	Monitoring Events .....	7
1.1.4	System .....	7
2	Terms and Definitions .....	8
2.1	Definitions .....	8
2.2	Abbreviations .....	8
3	Test Overview .....	9
3.1	Test Setup .....	9
3.1.1	Network Configuration for DUT .....	9
3.2	Prerequisites .....	10
3.3	Test Policy .....	10
3.3.1	General .....	10
3.3.2	Default Access Policy .....	11
3.3.3	Monitoring Events .....	12
3.3.4	System .....	13
4	Quick Install Test Cases .....	14
4.1	General .....	14
4.1.1	Factory default state verification .....	14
4.2	Default access policy .....	22
4.2.1	Default access policy - Anonymous .....	22
4.2.2	Default access policy - User .....	24
4.2.3	Default access policy - Administrator and Anonymous .....	29
4.2.4	Default access policy - Administrator and User/Operator .....	32
4.3	Monitoring Events .....	39
4.3.1	Processor Usage event .....	39
4.3.2	Last Reset event .....	42
4.3.3	Last Reboot event .....	46
4.3.4	Last Reboot event (Status change) .....	49
4.3.5	Last Clock Synchronization event .....	55
4.3.6	Last Clock Synchronization change event (SetSystemDateAndTime) .....	58
4.3.7	Last Clock Synchronization change event (NTP message) .....	64
4.3.8	Last Backup event .....	70
4.3.9	Fan Failure event .....	73
4.3.10	Power Supply Failure event .....	77



4.3.11	Storage Failure event .....	81
4.3.12	Critical Temperature event .....	84
4.4	System.....	88
4.4.1	Firmware Upload .....	88
4.4.2	Invalid Firmware Upload.....	93
Annex A	.....	97
A.1	Create user with defined user level.....	97
A.2	Restoring System Date and Time .....	101
A.3	Set NTP settings .....	103
A.4	Restoring NTP settings .....	104
A.5	Get service capabilities .....	105
A.6	Time synchronization .....	106



## 1 Introduction

The goal of the ONVIF test specification set is to make it possible to realize fully interoperable IP physical security implementation from different vendors. The set of ONVIF test specification describes the test cases needed to verify the [ONVIF Core Specs] and [ONVIF Conformance] requirements. In addition, the test cases are to be basic inputs for some Profile specification requirements. It also describes the test framework, test setup, pre-requisites, test policies needed for the execution of the described test cases.

This ONVIF Profile Q Test Specification acts as a supplementary document to the [ONVIF Core Specs], illustrating test cases need to be executed and passed. In addition, this specification acts as an input document to the development of test tool that will be used to test the ONVIF device implementation conformance towards ONVIF standard. This test tool is referred as ONVIF Client hereafter.

### 1.1 Scope

This ONVIF Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant devices. Conformance testing is meant to be functional black-box testing. The objective of this specification is to provide the test cases to test individual requirements of ONVIF devices according to ONVIF core services which are defined in [ONVIF Network Interface Specs].

The principal intended purposes are:

1. To provide self-assessment tool for implementations.
2. To provide comprehensive test suite coverage for [ONVIF Network Interface Specs].

This 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. Network protocol implementation Conformance test for HTTPS, HTTP, RTP and RTSP protocols.
4. Wi-Fi Conformance test.

The set of ONVIF Test Specification will not cover the complete set of requirements as defined in [ONVIF Network Interface Specs]; instead it will cover its subset.

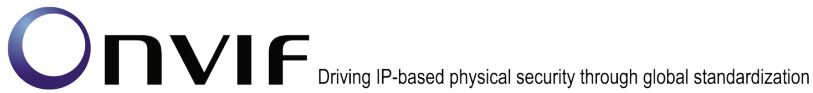
This ONVIF Profile Q Test Specification covers the Profile Q conformant transition from a device (unsecure) status after a factory reset command has been executed up to ONVIF Default Access Policy is applied. The following sections describe the brief overview and scope of each functional block.

#### 1.1.1 General

The General section covers the test cases needed for checking of Factory Default state and Operational state of the device.

The scope of this specification section is to cover the following functions:

- Turn device to Factory Default state with SetSystemFactoryDefault command
- Turn device to Operational state with creation of user with Administrator user level by



CreateUsers and SetUser command

- Check scopes Factory Default state and Operational state

### 1.1.2 Default Access Policy

The Default Access Policy section covers the test cases needed for checking of operations in the PRE\_AUTH, READ\_SYSTEM, READ\_MEDIA, ACTUATE, UNRECOVERABLE, WRITE\_SYSTEM and READ\_SYSTEM\_SECRET access classes.

### 1.1.3 Monitoring Events

Monitoring events test cases cover verification of monitoring events provided by ONVIF Core specification. It means that the following events are covered by these test cases:

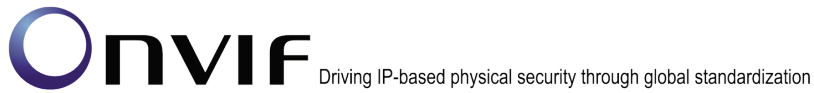
- tns1:Monitoring/ProcessorUsage
- tns1:Monitoring/OperatingTime/LastReset
- tns1:Monitoring/OperatingTime/LastReboot
- tns1:Monitoring/OperatingTime/LastClockSynchronization
- tns1:Monitoring/Backup/Last
- tns1:Device/HardwareFailure/FanFailure
- tns1:Device/HardwareFailure/PowerSupplyFailure
- tns1:Device/HardwareFailure/StorageFailure
- tns1:Device/HardwareFailure/TemperatureCritical

### 1.1.4 System

The System section covers the test cases needed for ONVIF device firmware upgrade via HTTP.

The scope of this specification section is to cover the following functions:

- Upgrade of firmware with **StartFirmwareUpgrade** command
- Check ONVIF device state after firmware upgrade.



## 2 Terms and Definitions

### 2.1 Definitions

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

### 2.2 Abbreviations

This section describes abbreviations used in this document.

<b>DUT</b>	Device Under Test
<b>HTTP</b>	Hyper Text Transport Protocol
<b>RTCP</b>	RTP Control Protocol
<b>RTSP</b>	Real Time Streaming Protocol
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>IP</b>	Internet Protocol
<b>IPv4</b>	Internet Protocol version 4
<b>NTP</b>	Network Time Protocol
<b>UTC</b>	Coordinated Universal Time



### 3 Test Overview

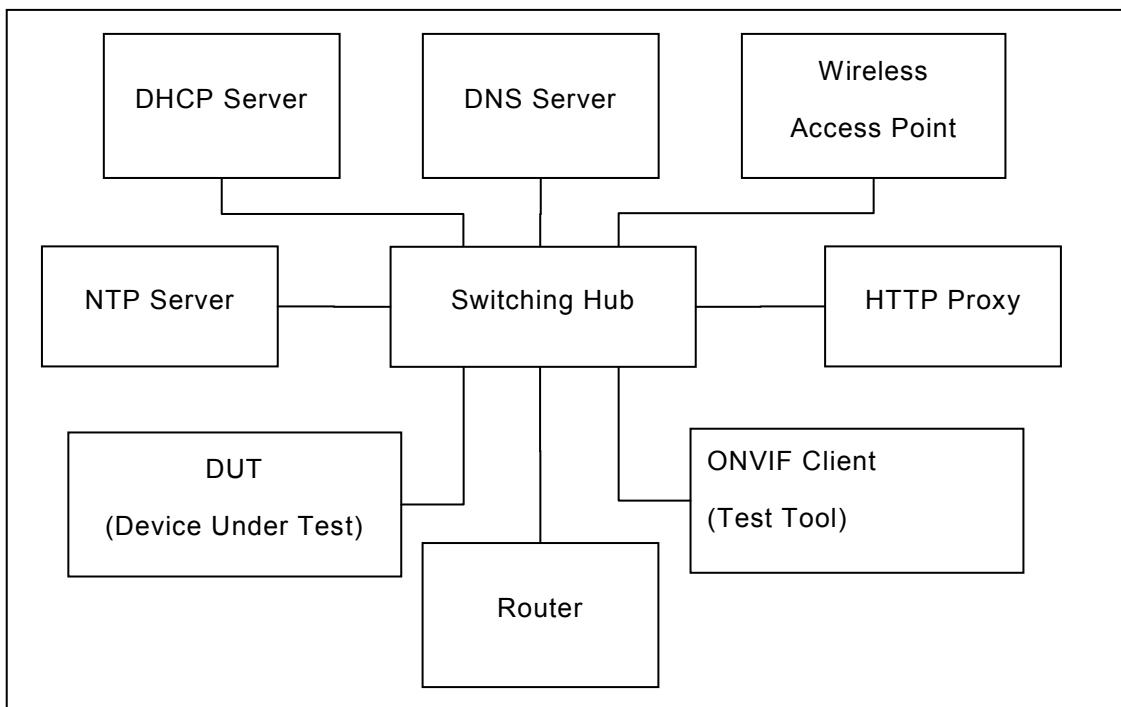
This section provides information the test setup procedure and required prerequisites, and the test policies that should be followed for test case execution.

#### 3.1 Test Setup

##### 3.1.1 Network Configuration for DUT

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

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



**Figure 1: Test Configuration for DUT**

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

**ONVIF Client (Test Tool):** Tests are executed by this system, and it controls the behaviour of the DUT. It handles both expected and unexpected behaviour.

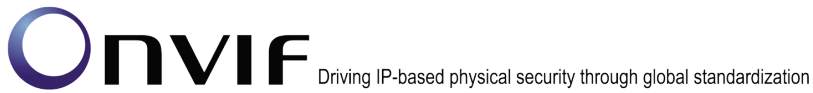
**HTTP Proxy:** provides facilitation in case of RTP and RTSP tunnelling over HTTP.

**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 ONVIF Client and DUT.



### 3.2 Prerequisites

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

- The DUT shall be in out-of-the-box state.

Test Operator shall configure Operation Delay properly so that it would have enough time to receive Notification messages for the following test cases for ONVIF Device Test Tool (see test description for more details):

- 4.3.1 Processor Usage event
- 4.3.2 Last Reset event
- 4.3.3 Last Reboot event
- 4.3.4 Last Reboot event (Status change)
- 4.3.5 Last Clock Synchronization event
- 4.3.6 Last Clock Synchronization change event (SetSystemDateAndTime)
- 4.3.7 Last Clock Synchronization change event (NTP message)
- 4.3.8 Last Backup event
- 4.3.9 Fan Failure event
- 4.3.10 Power Supply Failure event
- 4.3.11 Storage Failure event
- 4.3.12 Critical Temperature event

Test Operator shall configure Reboot Timeout properly so that it would have enough time to reboot device for the following test cases for ONVIF Device Test Tool (see test description for more details):

- 4.1.1 Factory default state verification
- **Ошибка! Источник ссылки не найден. Ошибка! Источник ссылки не найден.**
- 4.3.4 Last Reboot event (Status change)
- 4.4.1 Firmware Upload

### 3.3 Test Policy

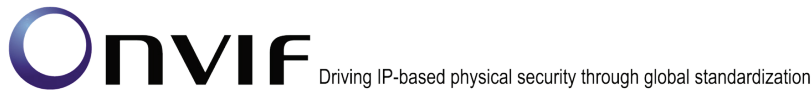
This section describes the test policies specific to the test case execution of each functional block.

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

#### 3.3.1 General

The DUT shall give the DeviceService entry point and EventService entry point by GetServices command. Otherwise these test cases will be skipped.

- DUT shall support the following commands:



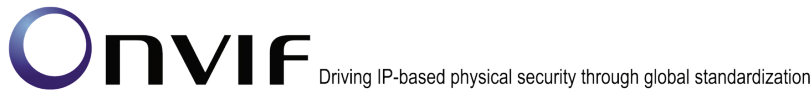
- GetServices
- GetServiceCapabilities
- DUT shall support HTTP Digest authentication
- DUT shall support set of user name with user name length equals to MaxUsernameLength
- DUT shall support set of user password with user password length equals to MaxPasswordLength
- DUT shall allow full anonymous access after hard SystemFactoryDefault
- DUT shall have onvif://www.onvif.org/Profile/Q/FactoryDefault scope after hard SystemFactoryDefault
- DUT shall not have onvif://www.onvif.org/Profile/Q/Operational scope after hard SystemFactoryDefault
- DUT shall have IPv4 network interface on DHCP after hard SystemFactoryDefault
- ZeroConfiguration shall be set to true after hard SystemFactoryDefault
- DUT shall conform to the access policy which is specified by Default Access Policy after create or set user with Administrator user level
- DUT shall have onvif://www.onvif.org/Profile/Q/Operational scope create or set user with Administrator user level
- DUT shall not have onvif://www.onvif.org/Profile/Q/FactoryDefault scope after create or set user with Administrator user level
- The following tests are performed
  - Quick Install SetSystemFactoryDefault state verification

Please refer to Section 4.1 for General Test Cases.

### 3.3.2 Default Access Policy

The Default Access Policy section covers the test cases needed for check of access policy on the DUT.

- The DUT shall support Default Access Policy
- DUT shall conform to the access policy which is specified by Default Access Policy
- The DUT shall support HTTP Digest Authentication
- The following tests are performed
  - Default Access Policy – Anonymous
  - Default Access Policy – User
  - Default Access Policy – Operator



- Default access policy - Administrator and Anonymous
- Default access policy - Administrator and User/Operator

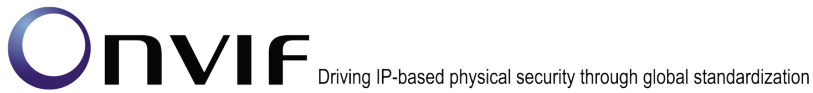
Please refer to Section 4.2 for Default Access Policy Test Cases.

### 3.3.3 Monitoring Events

The Monitoring Events section covers the test cases needed for check of monitoring property events.

DUT shall give the Event Service entry point by GetServices command. Otherwise these test cases will be skipped.

- DUT shall support following property events:
  - tns1:Monitoring/ProcessorUsage
  - tns1:Monitoring/OperatingTime/LastReset
  - tns1:Monitoring/OperatingTime/LastReboot
  - tns1:Monitoring/OperatingTime/LastClockSynchronization
  - tns1:Monitoring/Backup/Last
  - tns1:Device/HardwareFailure/FanFailure
  - tns1:Device/HardwareFailure/PowerSupplyFailure
  - tns1:Device/HardwareFailure/StorageFailure
  - tns1:Device/HardwareFailure/TemperatureCritical
- DUT shall support GetEventProperties command and return all supported events in TopicSet.
- DUT shall support Pull Point Subscription and Topic Expression filter.
- DUT shall generate property events with initial state after subscription was done.
- DUT shall generate property events with current state after corresponding properties were changed.
- The following tests are performed
  - Getting of Processor Usage event and generate Processor Usage property event with initial state
  - Getting of Last Reset event and generate Last Reset property event with initial state
  - Getting of Last Reset event and generate Last Reboot property event with initial state
  - Generate Last Reboot property event after device reboot
  - Getting of Last Clock Synchronization event and generate Last Clock Synchronization property event with initial state



- Generate Last Clock Synchronization property event after clock synchronization via SetSystemDateAndTime command
- Generate Last Clock Synchronization property event after clock synchronization with NTP server
- Getting of Last Backup event and generate Last Backup property event with initial state
- Getting of Fan Failure event and generate Fan Failed property event with initial state
- Getting of Power Supply Failure event and generate Power Supply Failed property event with initial state
- Getting of Storage Failure event and generate Storage Failed property event with initial state
- Getting of Critical Temperature event and generate Critical Temperature property event with initial state

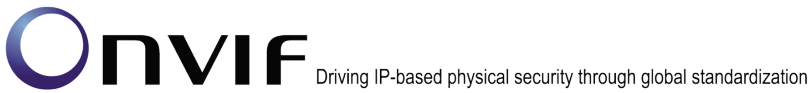
Please refer to Section 4.3 for Monitoring Events Test Cases.

#### **3.3.4 System**

The System section covers the test cases with firmware upgrade via HTTP.

- DUT shall support HttpFirmwareUpgrade.
- DUT shall returns HTTP 415 message in case invalid firmware file.
- The following tests are performed
  - Firmware Upgrade via HTTP with valid firmware file
  - Firmware Upgrade via HTTP with invalid firmware file

Please refer to Section 4.4 for System Test Cases.



## 4 Quick Install Test Cases

### 4.1 General

#### 4.1.1 Factory default state verification

**Test Label:** Quick Install SetSystemFactoryDefault state verification

**Test Case ID:** QUICK\_INSTALL-1-1-1

**ONVIF Core Specification Coverage:** Factory Default, CreateUsers (ONVIF Core Specification), SetUser (ONVIF Core Specification)

**Command Under Test:** SetSystemFactoryDefault, CreateUsers, SetUsers

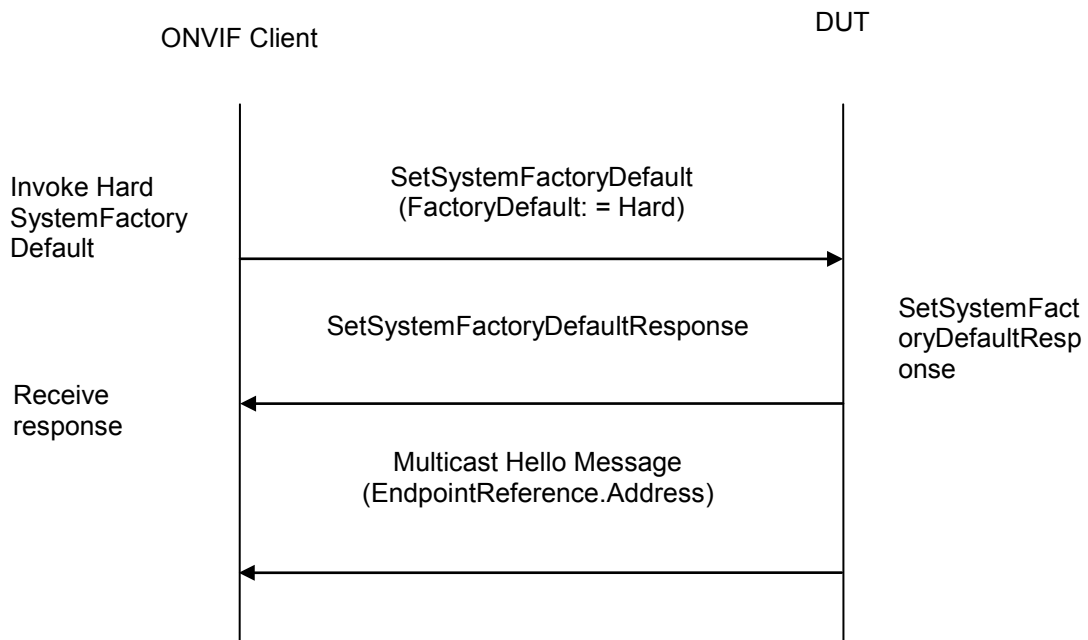
**WSDL Reference:** devicemgmt.wsdl

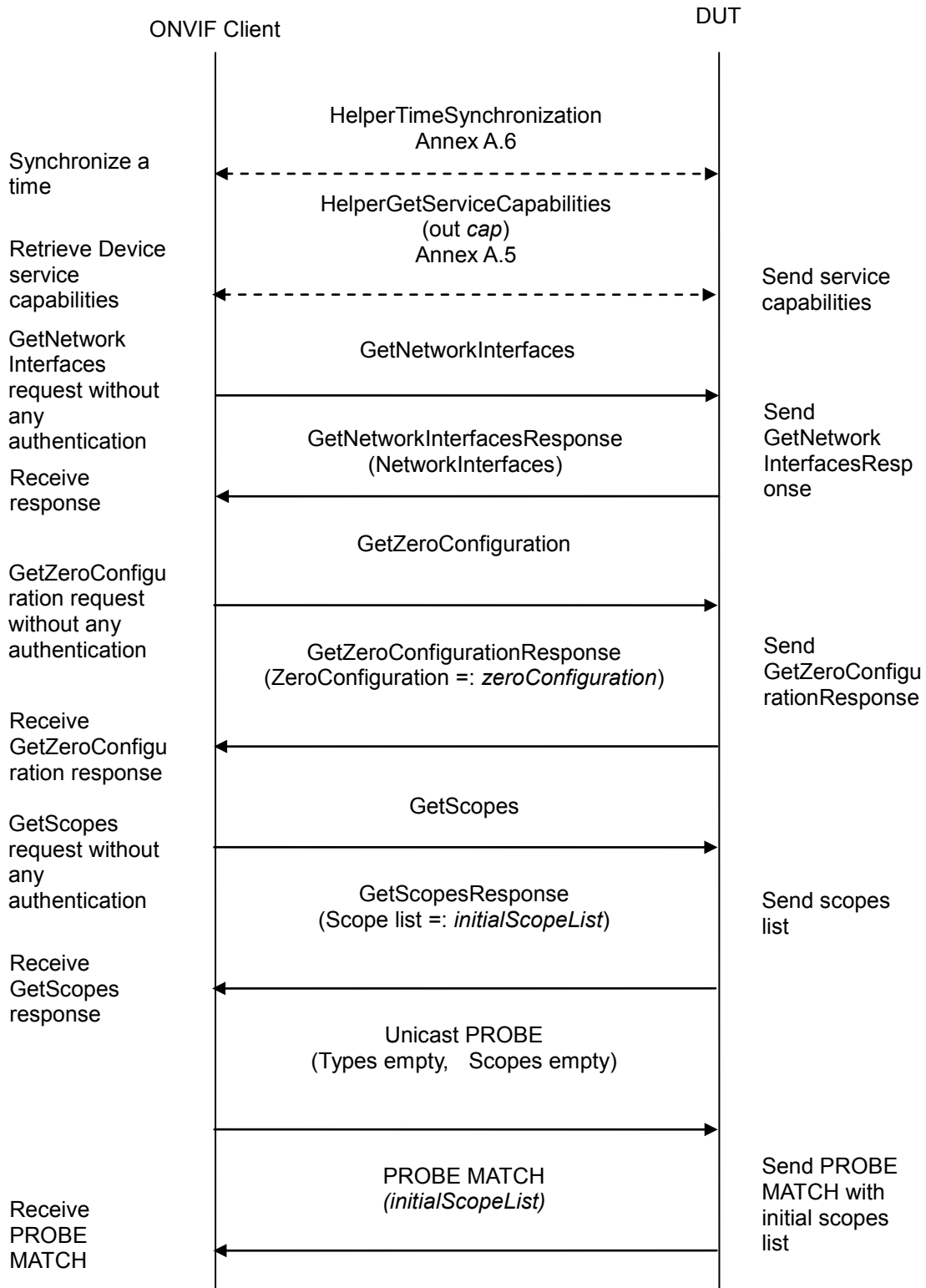
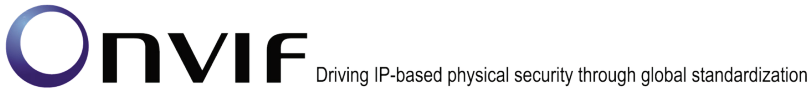
**Test Purpose:** To verify transition from FactoryDefault state to Operational state.

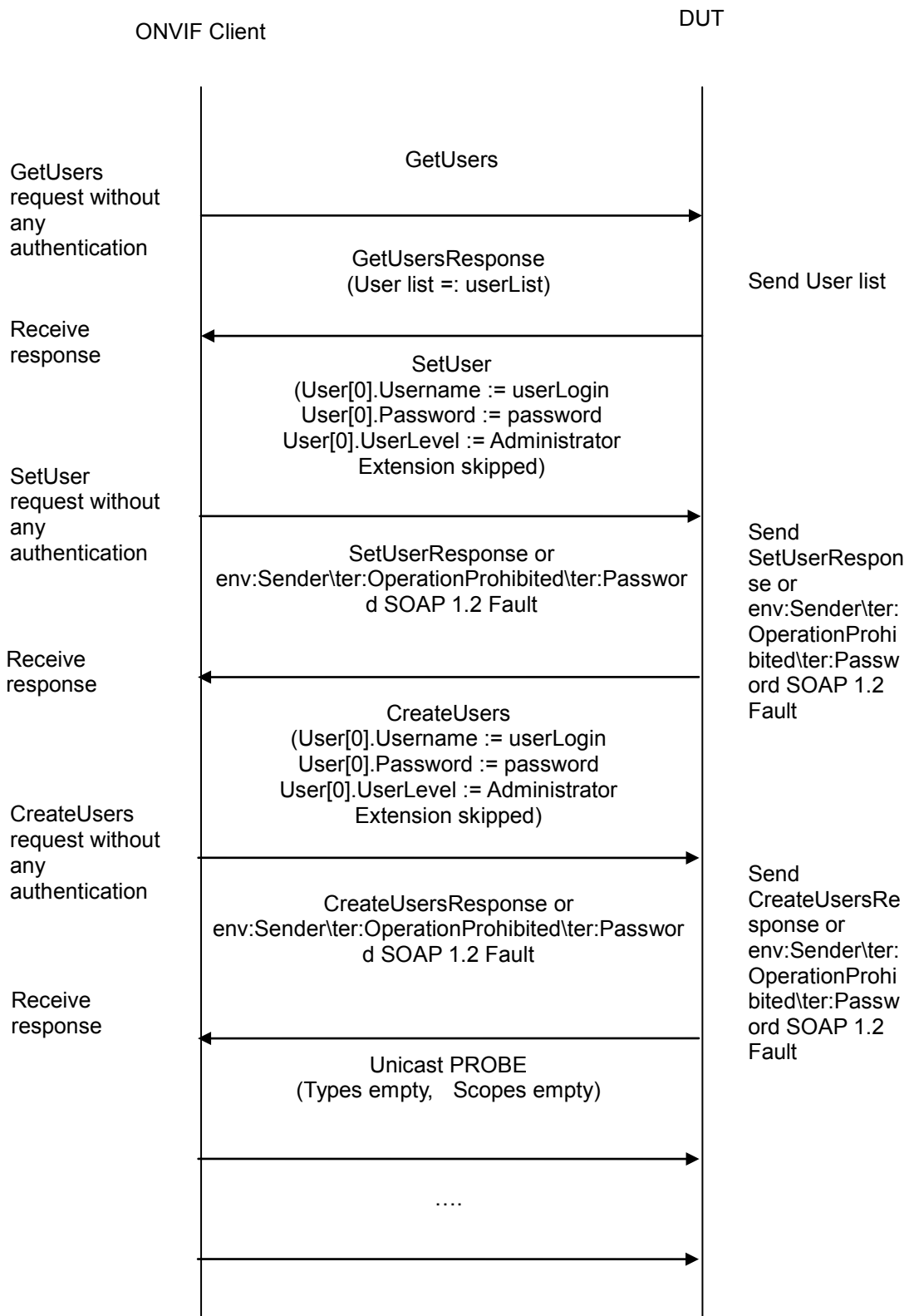
**Pre-requisite:** GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. ZeroConfiguration is supported by the DUT as indicated by the Network.ZeroConfiguration capability. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security.MaxUsernameLength, Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security.MaxPasswordLength.

**Test Configuration:** ONVIF Client and DUT

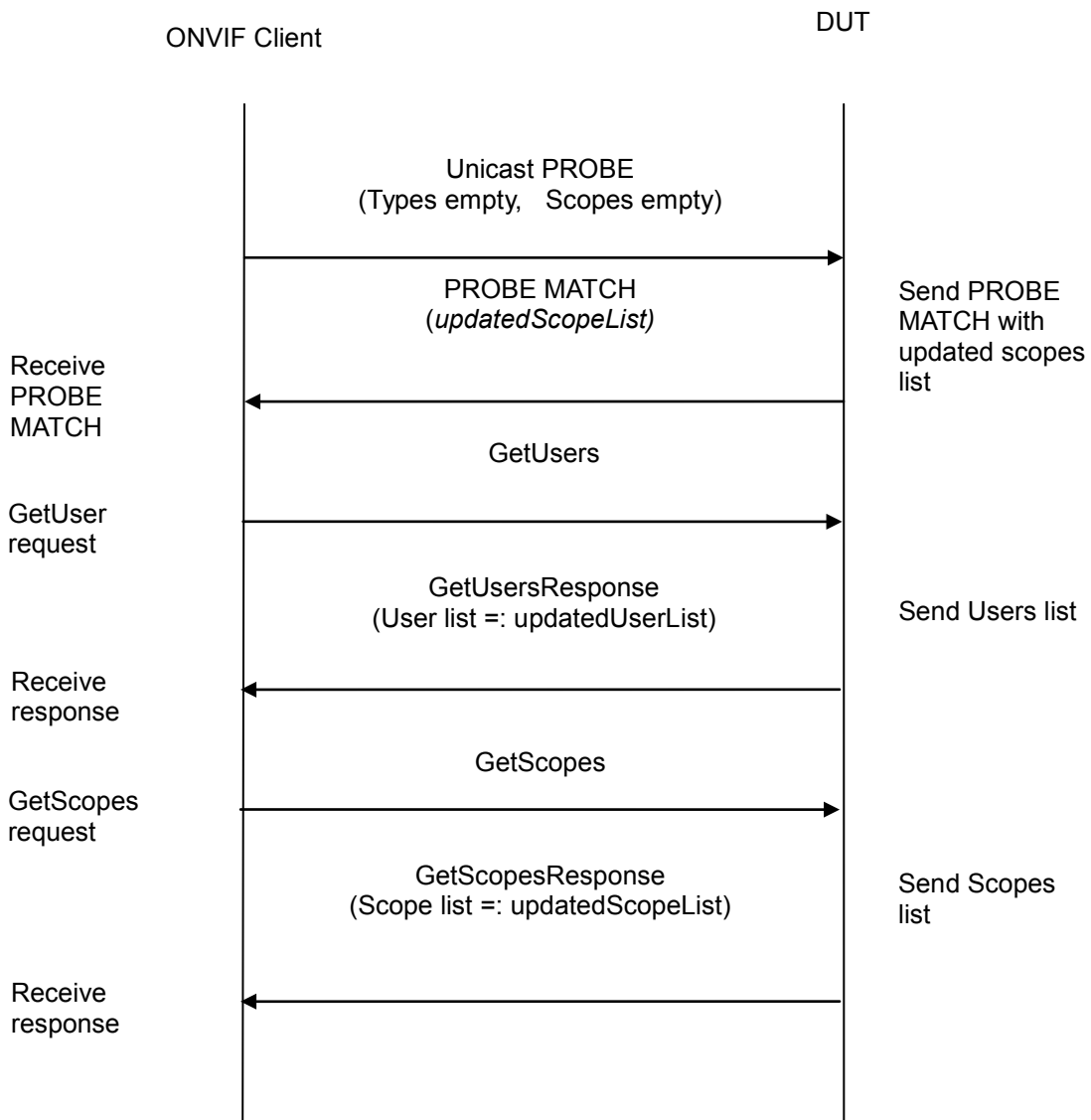
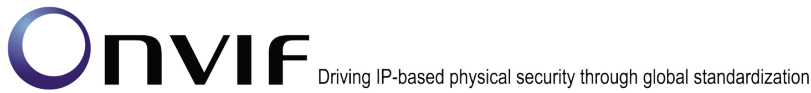
**Test Sequence:**





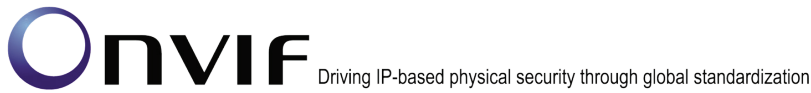




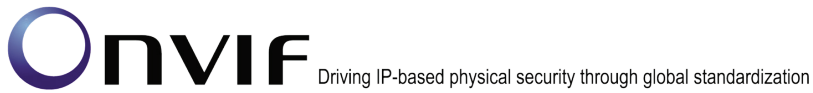


**Test Procedure:**

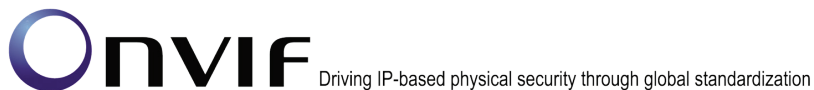
1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **SetSystemFactoryDefault** with parameters
  - FactoryDefault := Hard
4. DUT responds with a **SetSystemFactoryDefaultResponse** message.
5. Until *timeout1* timeout expires, repeat the following steps:
  - 5.1. The DUT will send Multicast **Hello** message after it is successfully rebooted with parameters:



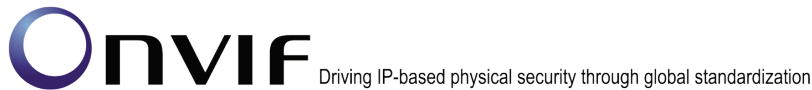
- EndpointReference.Address equal to unique endpoint reference of the DUT
  - Types list
  - Scopes list := *scopesList*
  - XAddrs list := *xaddrsList*
  - MetadataVersion
- 5.2. If *xaddrsList* contains URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, go to step 7.
6. If *timeout1* timeout expires for step 5 without Hello with URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, FAIL the test and skip other steps.
  7. ONVIF client waits for 5 seconds after Hello was received.
  8. ONVIF client checks the following:
    - If *scopesList* does not contain “ onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.
    - If *scopesList* contains “ onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test and skip other steps.
  9. ONVIF Client synchronize a time between ONVIF Client and DUT by following the procedure mentioned in Annex A.6.
  10. ONVIF Client gets the service capabilities without any authentication (out *cap*) by following the procedure mentioned in Annex A.5.
  11. ONVIF client invokes **GetNetworkInterfaces** without any authentication.
  12. The DUT responds with **GetNetworkInterfacesResponse** message with parameters
    - NetworkInterfaces.token1 =: *currentNetInt*, where token1 is token of the currently used network interface
  13. If *currentNetInt*.IPv4.DHCP is not equal to true, FAIL the test and skip other steps.
  14. If *cap* contains Network.IPVersion6 item and *cap*.Network.IPVersion6 equals to true:
    - If *currentNetInt*.IPv6.Enabled is not equal to true, FAIL the test and skip other steps.
    - If *currentNetInt*.IPv6.Config does not contain at least one LinkLocal item with not empty Address item, FAIL the test and skip other steps.
  15. ONVIF client invokes **GetZeroConfiguration** without any authentication.
  16. The DUT responds with **GetZeroConfigurationResponse** message with parameters
    - ZeroConfiguration =: *zeroConfiguration*
  17. If *zeroConfiguration*.Enabled is not equal to true, FAIL the test.
  18. ONVIF Client invokes **GetScopes** without any authentication.



19. DUT responds with a **GetScopesResponse** message with parameters
  - Scope list =: *initialScopeList*
20. If *initialScopeList* does not contain “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test.
21. If *initialScopeList* contains “onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test.
22. ONVIF Client invokes Unicast **PROBE** message with the following parameters
  - Types empty
  - Scopes empty
23. DUT responds with a **PROBE MATCH** message with parameters:
  - ProbeMatch.Scopes
24. Set the following:
  - *initialScopeList* := ProbeMatch.Scopes
25. If *initialScopeList* does not contain “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.
26. If *initialScopeList* contains “onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test and skip other steps.
27. If *cap* does not contain Security.MaxPasswordLength or Security.MaxUserNameLength, FAIL the test and skip other steps.
28. ONVIF Client invokes **GetUsers** without any authentication.
29. DUT responds with a **GetUsersResponse** message with parameters.
  - User list =: *userList*
30. If *userList* contains user with user level Administrator:
  - 30.1. Set the following:
    - *passwordLength* := *cap*.Security.MaxPasswordLength
    - *userLogin* := Username of user with user level equal to Administrator from *userList*
    - *password* := random string, contains *passwordLength* ASCII characters
  - 30.2. ONVIF Client invokes **SetUser** with parameters
    - User[0].Username := *userLogin*
    - User[0].Password := *password*
    - User[0].UserLevel := Administrator
    - Extension skipped



- 30.3. If the DUT responds with **SetUserResponse** message, go to 32.
- 30.4. If DUT returns env:Sender\ter:OperationProhibited\ter>Password SOAP 1.2 fault:
  - 30.4.1. Set the following:
    - *password* := random string, contains *passwordLength* ASCII characters
  - 30.4.2. Go to the step 30.2.
- 30.5. If DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.
31. If *userList* does not contain user with user level Administrator:
  - 31.1. Set the following:
    - *userLoginLength* := *cap.Security.MaxUserNameLength*
    - *passwordLength* := *cap.Security.MaxPasswordLength*
    - *userLogin* := random string, contains *userLoginLength* low case alphabet characters, differs from usernames listed in *userList*
    - *password* := random string, contains *passwordLength* ASCII characters
  - 31.2. ONVIF Client invokes **CreateUsers** with parameters
    - User[0].Username := *userLogin*
    - User[0].Password := *password*
    - User[0].UserLevel := Administrator
    - Extension skipped
  - 31.3. If the DUT responds with **CreateUsersResponse** message, go to 32.
  - 31.4. If the DUT returns env:Sender\ter:OperationProhibited\ter>Password SOAP 1.2 fault:
    - 31.4.1. Set the following:
      - *password* := random string, contains *passwordLength* ASCII characters
    - 31.4.2. Go to the step 31.2.
    - 31.4.3. If DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.
32. ONVIF Client waits the Reboot Timeout.
33. ONVIF Client sends PROBE message and if DUT responds with PROBE MATCH message then go to the step 35
34. ONVIF Client waits for Hello message sent from newly configured address by the DUT. Then ONVIF Client starts using this newly configured address for further communications with DUT.
35. Until *timeout1* expires, repeat the following steps:
  - 35.1. ONVIF Client invokes Unicast **PROBE** message with the following parameters



- Types empty
- Scopes empty

35.2. If the DUT responds with **PROBE MATCH** message:

35.2.1. Set the following:

- *updatedScopeList* := ProbeMatch.Scopes

35.2.2. If *updatedScopeList* contains “onvif://www.onvif.org/Profile/Q/Operational” scope, go to the step 35.

35.3. If *timeout1* timeout expires for step 35 without **PROBE MATCH** message with “onvif://www.onvif.org/Profile/Q/Operational” scope in *updatedScopeList*, FAIL the test and skip other steps.

36. If *updatedScopeList* contains “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.

37. ONVIF Client invokes **GetUsers** with user with the user level Administrator credentials (*userLogin* and *password*).

38. DUT responds with a **GetUsersResponse** message with parameters.

- User list =: *updatedUserList*

39. If *updatedUserList* does not contain Username = *userLogin* with UserLevel = Administrator, FAIL the test and skip other steps.

40. ONVIF Client invokes **GetScopes** with the user level Administrator credentials (*userLoginUser* and *passwordUser*)

41. The DUT responds with a **GetScopesResponse** message with parameters

- Scope list =: *updatedScopeList*

42. If *updatedScopeList* contains “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.

43. If *updatedScopeList* does not contain “onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test and skip other steps.

#### Procedure Result:

##### PASS –

The DUT passed all assertions.

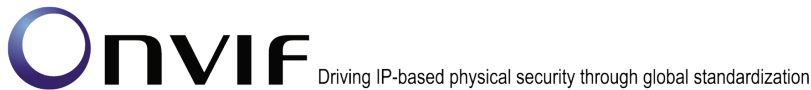
##### FAIL –

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

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

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

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



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

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

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

The DUT did not allow Anonymous access to the **GetNetworkInterfaces** command.

The DUT did not allow Anonymous access to the **GetZeroConfiguration** command.

The DUT did not allow Anonymous access to the **GetUsers** command before changing the user with the user level Administrator credentials.

The DUT did not allow Anonymous access to the **SetUsers** command before changing the user with the user level Administrator credentials.

The DUT did not allow Anonymous access to the **CreateUsers** command before changing the user with the user level Administrator credentials.

The DUT did not allow Anonymous access to the **GetScopes** command before changing the user with the user level Administrator credentials.

The DUT allowed Anonymous access to the **GetUsers** command after changing the user with the user level Administrator credentials.

The DUT allowed Anonymous access to the **GetScopes** command after changing the user with the user level Administrator credentials.

**Note:** User with username *userLogin* and password *password* shall be used for further test cases.

**Note:** *timeout1* will be taken from Reboot Timeout field of ONVIF Device Test Tool.

**Note:** IPv4 address from Hello shall be used for further test cases.

## **4.2 Default access policy**

### **4.2.1 Default access policy - Anonymous**

**Test Label:** Default Access Policy - Anonymous

**Test Case ID:** QUICK\_INSTALL-2-1-1

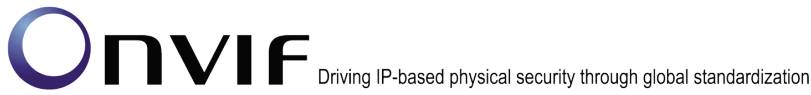
**ONVIF Core Specification Coverage:** Default access policy

**Command Under Test:** GetServices, GetServiceCapabilities, GetHostname, GetSystemDateAndTime

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify that operations in the PRE\_AUTH access class can be accessed without authentication being required.

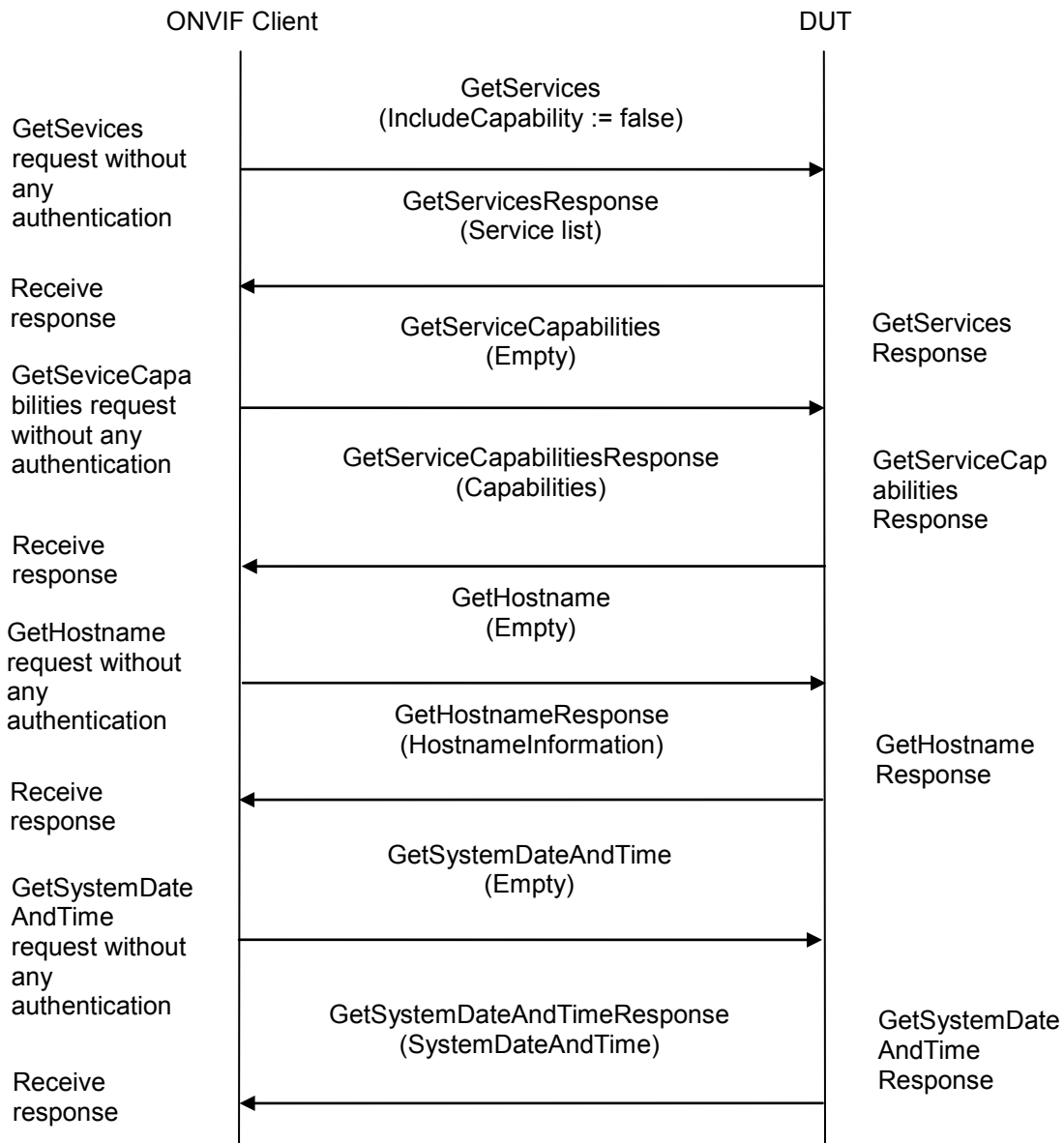
**Pre-requisite:** GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication



is supported by the DUT as indicated by the Security.HttpDigest capability. Default Access Policy is not modified.

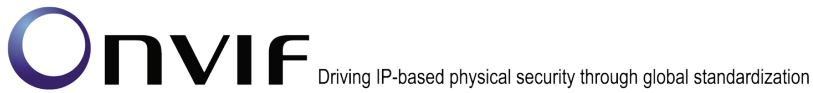
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF client invokes **GetServices** for Device service without any authentication with parameters



- IncludeCapability := false
4. The DUT responds with **GetServicesResponse** message with parameters
    - Service list
  5. ONVIF client invokes **GetServiceCapabilities** for Device Service without any authentication.
  6. The DUT responds with **GetServiceCapabilitiesResponse** message with parameters
    - Capabilities
  7. ONVIF client invokes **GetHostname** without any authentication.
  8. The DUT responds with **GetHostnameResponse** message with parameters
    - HostnameInformation
  9. ONVIF client invokes **GetSystemDateAndTime** without any authentication.
  10. The DUT responds with **GetSystemDateAndTimeResponse** message with parameters
    - SystemDateAndTime

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

The DUT did not allow Anonymous access to the **GetServices** command.

The DUT did not allow Anonymous access to the **GetServiceCapabilities** command.

The DUT did not allow Anonymous access to the **GetHostname** command.

The DUT did not allow Anonymous access to the **GetSystemDateAndTime** command.

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

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

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

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

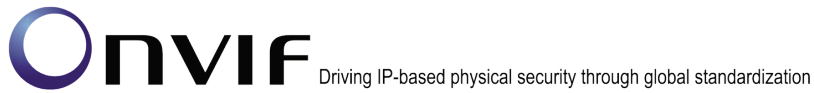
**4.2.2 Default access policy - User**

**Test Label:** Default Access Policy - User

**Test Case ID:** QUICK\_INSTALL-2-1-2

**ONVIF Core Specification Coverage:** Default access policy





**Command Under Test:** GetNTP, GetNetworkInterfaces, GetScopes, GetDiscoveryMode, GetEventProperties

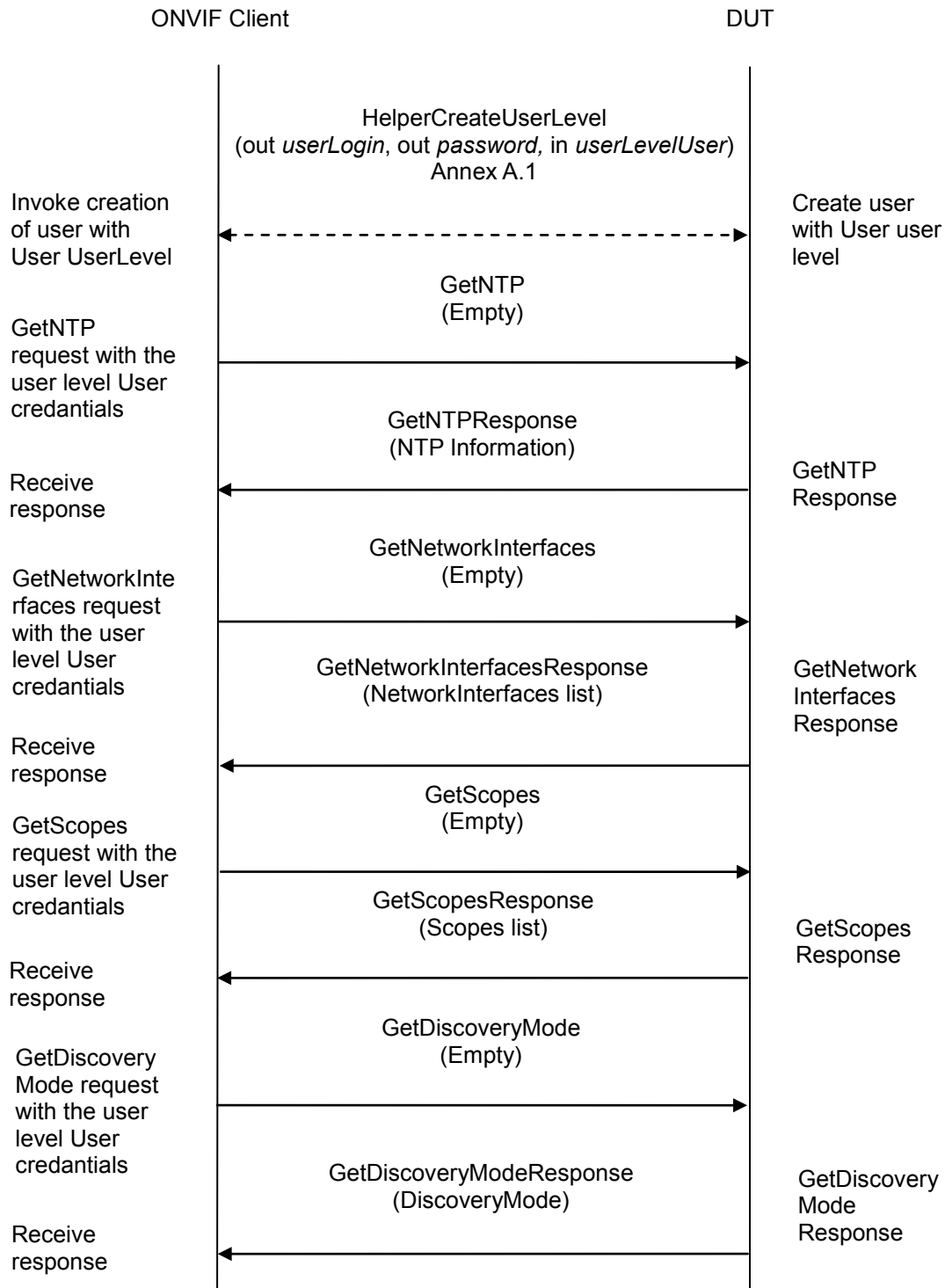
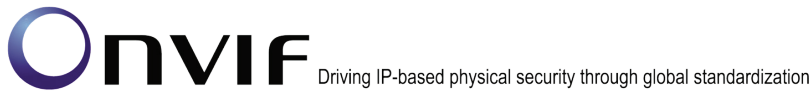
**WSDL Reference:** devicemgmt.wsdl

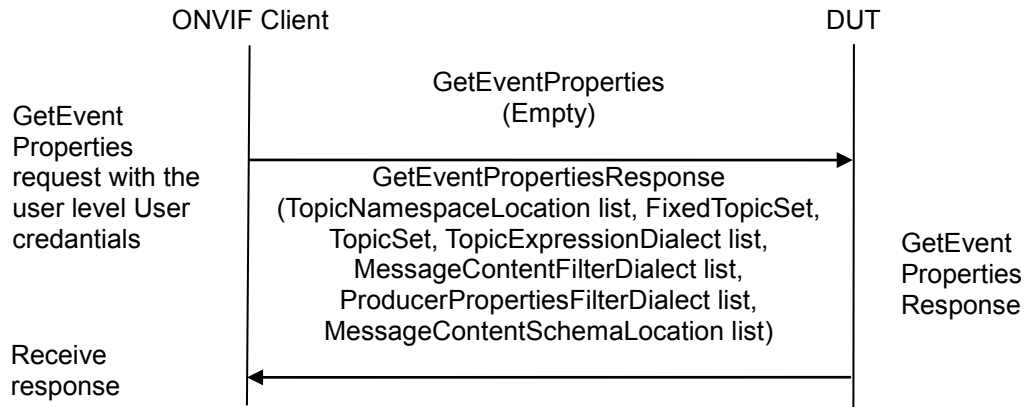
**Test Purpose:** To verify that operations in the READ\_SYSTEM and READ\_MEDIA access classes can be accessed with authentication level User.

**Pre-requisite:** GetServices command is supported by the DUT. Event Service was received from the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security.MaxUsernameLength, Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security.MaxPasswordLength. Default Access Policy is not modified.

**Test Configuration:** ONVIF Client and DUT

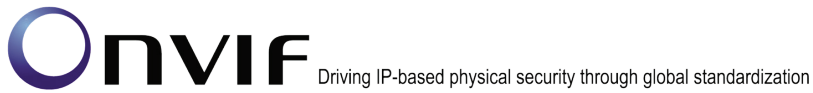
**Test Sequence:**





### Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. Set the following:
  - *userLevel* := User
4. ONVIF Client generates creates user with predefined user level (in *userLevel*) and user login (out *userLogin*) and password (out *password*) by following the procedure mentioned in Annex A.1.
5. If the DUT supports NTP as indicated by Network.NTP capability:
  - 5.1. ONVIF client invokes **GetNTP** without any authentication.
  - 5.2. The DUT responds with **HTTP 401 Unauthorized** error.
  - 5.3. ONVIF client invokes **GetNTP** with user with the user level User credentials (*userLogin* and *password*).
  - 5.4. The DUT responds with **GetNTPResponse** message with parameters
    - NTPInformation
6. ONVIF client invokes **GetNetworkInterfaces** without any authentication.
7. The DUT responds with **HTTP 401 Unauthorized** error.
8. ONVIF client invokes **GetNetworkInterfaces** with user with the user level User credentials (*userLogin* and *password*).
9. The DUT responds with **GetNetworkInterfacesResponse** message with parameters
  - NetworkInterfaces list



10. ONVIF client invokes **GetScopes** without any authentication.
11. The DUT responds with **HTTP 401 Unauthorized** error.
12. ONVIF client invokes **GetScopes** with user with the user level User credentials (*userLogin* and *password*).
13. The DUT responds with **GetScopesResponse** message with parameters
  - Scopes list
14. ONVIF client invokes **GetDiscoveryMode** without any authentication.
15. The DUT responds with **HTTP 401 Unauthorized** error.
16. ONVIF client invokes **GetDiscoveryMode** with user with the user level User credentials (*userLogin* and *password*).
17. The DUT responds with **GetDiscoveryModeResponse** message with parameters
  - DiscoveryMode
18. ONVIF client invokes **GetEventProperties** without any authentication.
19. The DUT responds with **HTTP 401 Unauthorized** error.
20. ONVIF client invokes **GetEventProperties** with user with the user level User credentials (*userLogin* and *password*).
21. The DUT responds with **GetEventPropertiesResponse** message with parameters
  - TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - ProducerPropertiesFilterDialect list
  - MessageContentSchemaLocation list
  - Other additional fields if any

**Test Result:**

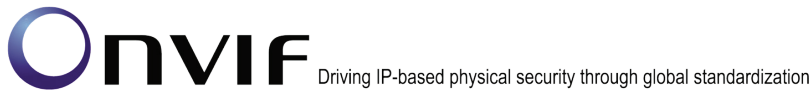
**PASS –**

The DUT passed all assertions.

**FAIL –**

The DUT allow Anonymous access to the **GetNTP** command.

The DUT allow Anonymous access to the **GetNetworkInterfaces** command.



The DUT allow Anonymous access to the **GetScopes** command.

The DUT allow Anonymous access to the **GetDiscoveryMode** command.

The DUT allow Anonymous access to the **GetEventProperties** command.

The DUT did not allow user with the user level User access to the **GetNTP** command.

The DUT did not allow user with the user level User access to the **GetNetworkInterfaces** command.

The DUT did not allow user with the user level User access to the **GetScopes** command.

The DUT did not allow user with the user level User access to the **GetDiscoveryMode** command.

The DUT did not allow user with the user level User access to the **GetEventProperties** command.

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

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

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

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

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

#### 4.2.3 Default access policy - Administrator and Anonymous

**Test Label:** Default Access Policy - Administrator And Anonymous

**Test Case ID:** QUICK\_INSTALL-2-1-4

**ONVIF Core Specification Coverage:** Default access policy

**Command Under Test:** SetScopes, SetDiscoveryMode, GetAccessPolicy, CreateUsers, SetSystemDateAndTime

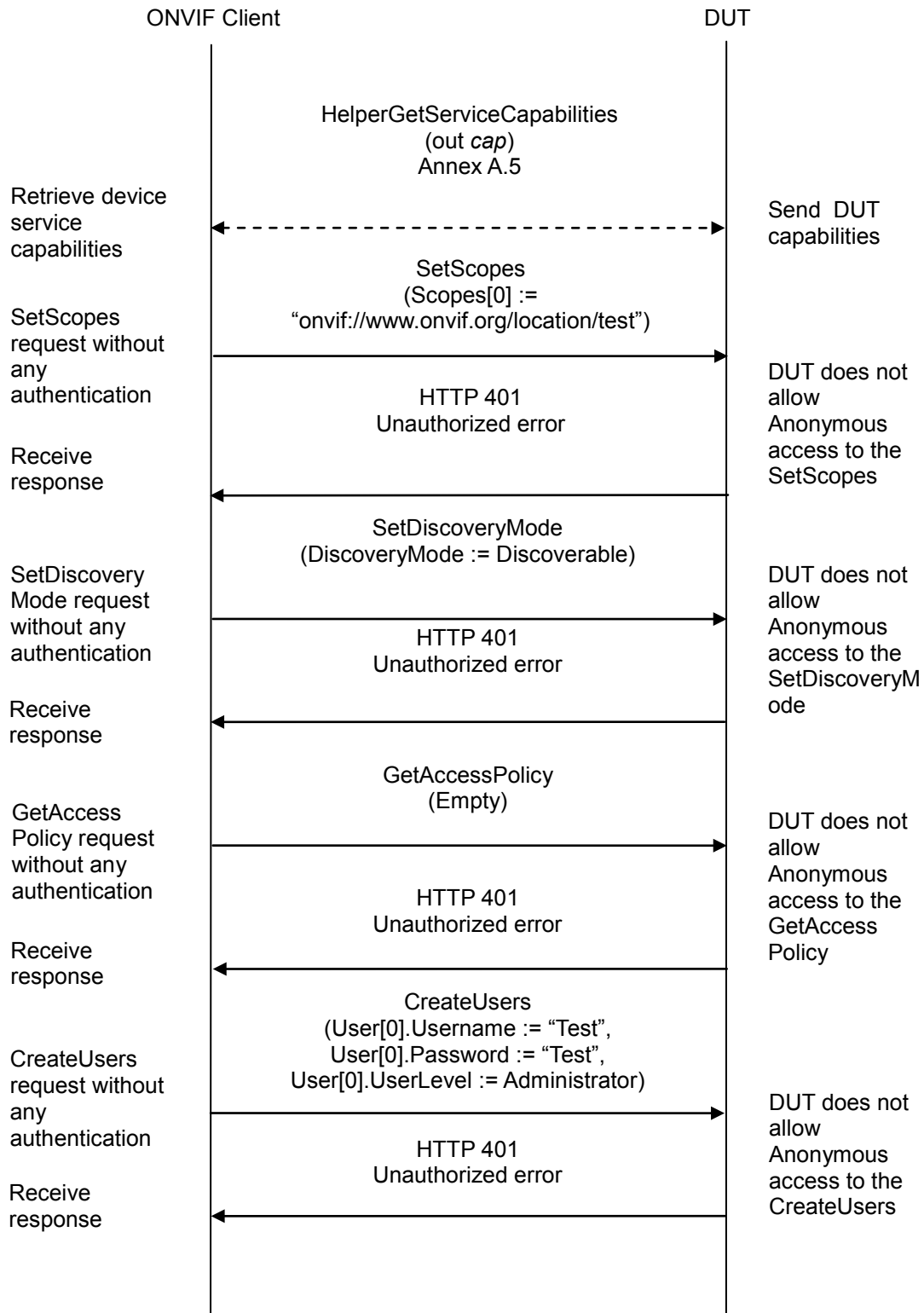
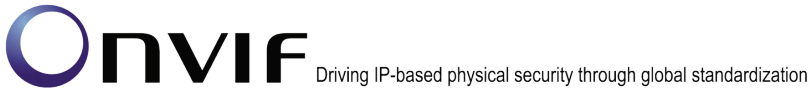
**WSDL Reference:** devicemgmt.wsdl

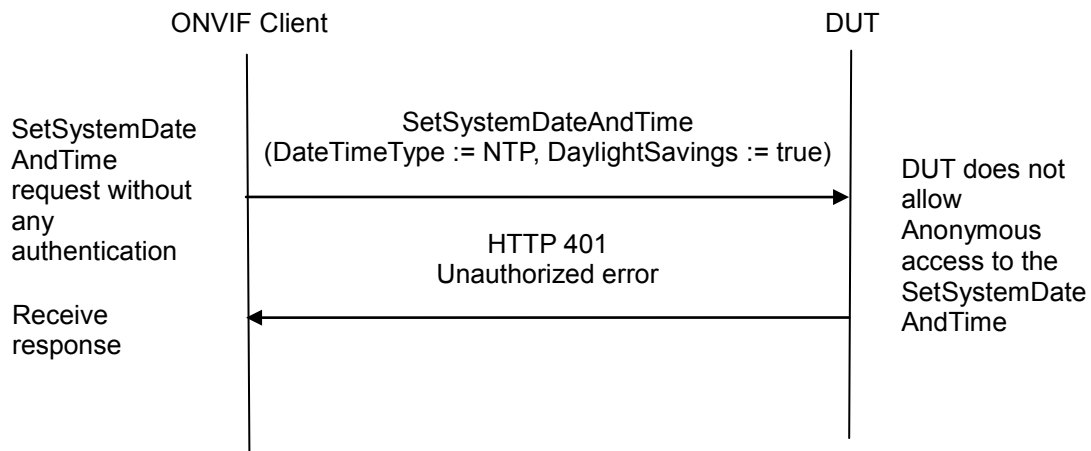
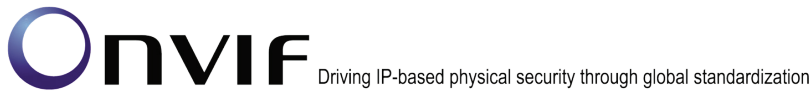
**Test Purpose:** To verify that operations in the UNRECOVERABLE, WRITE\_SYSTEM and READ\_SYSTEM\_SECRET access classes can not be accessed without authentication level Administrator.

**Pre-requisite:** GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. Default Access Policy is not modified.

**Test Configuration:** ONVIF Client and DUT

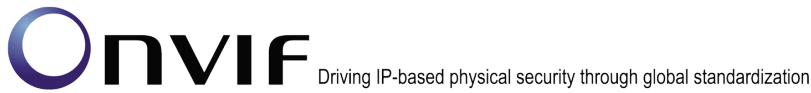
**Test Sequence:**





#### Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client gets the service capabilities (out *cap*) by following the procedure mentioned in Annex A.5.
4. ONVIF client invokes **SetScopes** without any authentication with parameters
  - Scopes[0] := "onvif://www.onvif.org/location/test"
5. The DUT responds with **HTTP 401 Unauthorized** error.
6. ONVIF client invokes **SetDiscoveryMode** without any authentication with parameters
  - DiscoveryMode := Discoverable
7. The DUT responds with **HTTP 401 Unauthorized** error.
8. If *cap.Security* contains *AccessPolicyConfig* and *cap.Security.AccessPolicyConfig* equals to true:
  - 8.1. ONVIF client invokes **GetAccessPolicy** without any authentication.
  - 8.2. The DUT responds with **HTTP 401 Unauthorized** error.
9. ONVIF client invokes **CreateUsers** without any authentication with parameters
  - User[0].Username := "Test"
  - User[0].Password := "Test"
  - User[0].UserLevel := Administrator
  - Extension skipped
10. The DUT responds with **HTTP 401 Unauthorized** error.



11. ONVIF client invokes **SetSystemDateAndTime** without any authentication with parameters

- DateTimeType := NTP
- DaylightSavings := true
- TimeZone skipped
- UTCDateTime skipped

12. The DUT responds with **HTTP 401 Unauthorized** error.

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

The DUT allowed Anonymous access to the **SetScopes** command.

The DUT allowed Anonymous access to the **SetDiscoveryMode** command.

The DUT allowed Anonymous access to the **GetAccessPolicy** command.

The DUT allowed Anonymous access to the **CreateUsers** command.

The DUT allowed Anonymous access to the **SetSystemDateAndTime** command.

#### 4.2.4 Default access policy - Administrator and User/Operator

**Test Label:** Default Access Policy - Administrator And User/Operator

**Test Case ID:** QUICK\_INSTALL-2-1-5

**ONVIF Core Specification Coverage:** Default access policy

**Command Under Test:** SetScopes, SetDiscoveryMode, GetAccessPolicy, CreateUsers, SetSystemDateAndTime

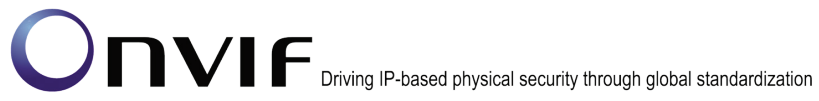
**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify that operations in the UNRECOVERABLE, WRITE\_SYSTEM and READ\_SYSTEM\_SECRET access classes can not be accessed without authentication level Administrator.

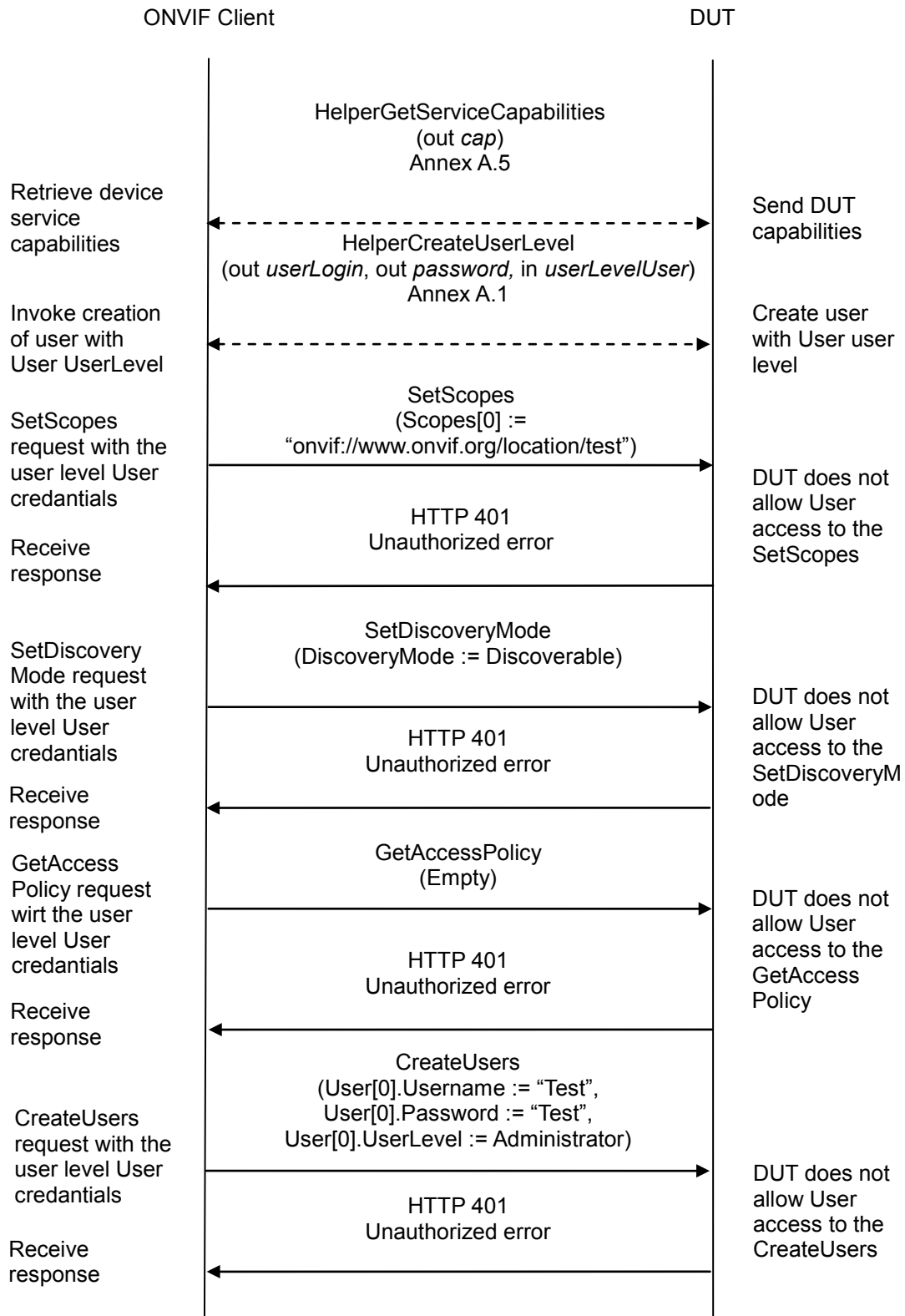
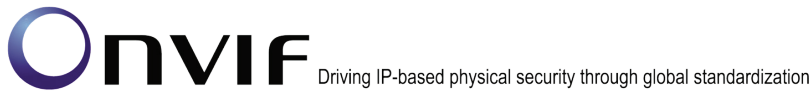
**Pre-requisite:** GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security.MaxUsernameLength, Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security.MaxPasswordLength. Default Access Policy is not modified.

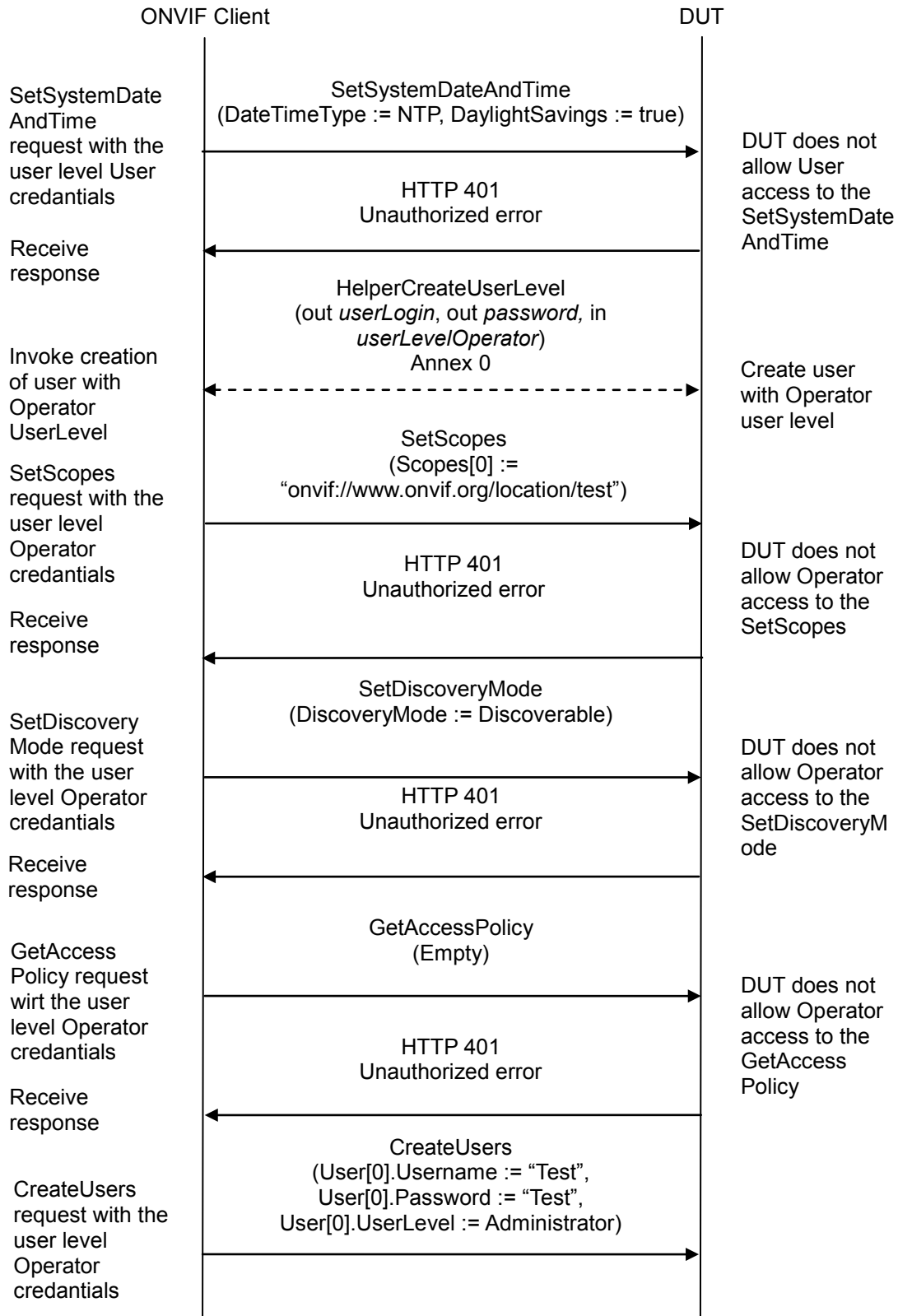
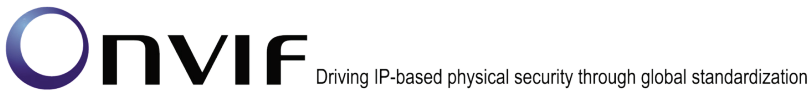
**Test Configuration:** ONVIF Client and DUT

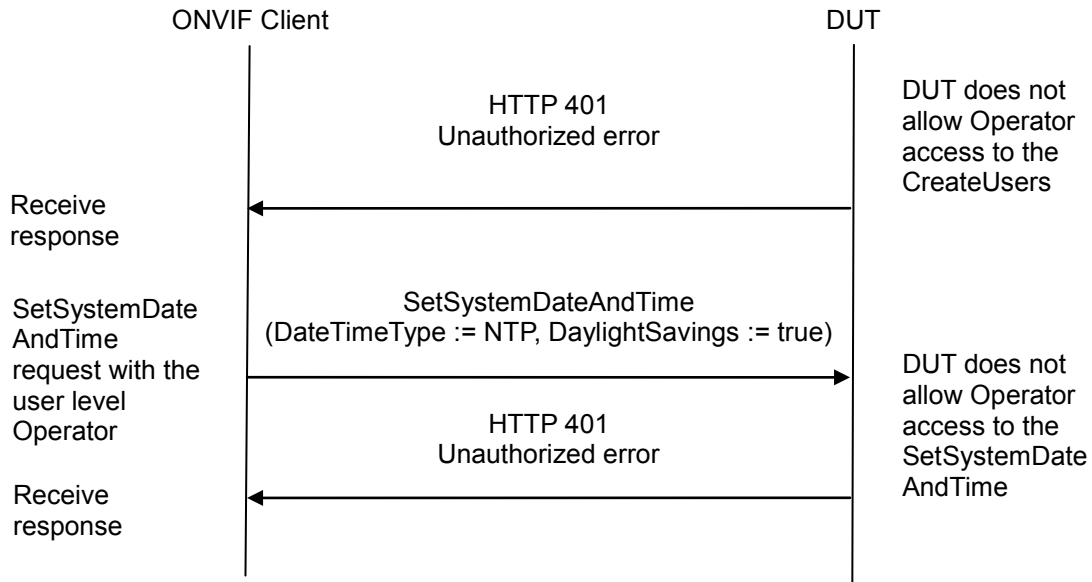
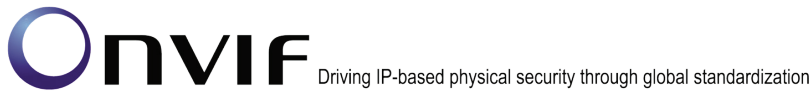




**Test Sequence:**

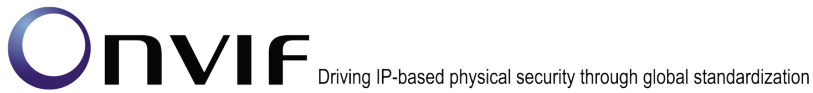




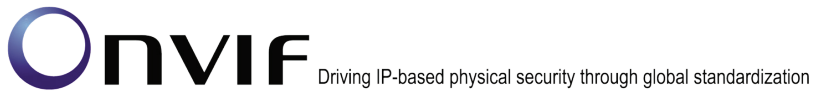


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. Set the following:
  - *userLevelUser* := User
  - *userLevelOperator* := Operator
4. ONVIF Client generates creates user with predefined user level (in *userLevelUser*) and user login (out *userLoginUser*) and password (out *passwordUser*) by following the procedure mentioned in Annex A.1.
5. ONVIF Client gets the service capabilities (out *cap*) by following the procedure mentioned in Annex A.5.
6. ONVIF client invokes **SetScopes** with user with the user level User credentials (*userLoginUser* and *passwordUser*) with parameters
  - *Scopes*[0] := "onvif://www.onvif.org/location/test"
7. The DUT responds with **HTTP 401 Unauthorized** error.
8. ONVIF client invokes **SetDiscoveryMode** with user with the user level User credentials (*userLoginUser* and *passwordUser*) with parameters
  - *DiscoveryMode* := Discoverable



9. The DUT responds with **HTTP 401 Unauthorized** error.
10. If *cap.Security* contains *AccessPolicyConfig* and *cap.Security.AccessPolicyConfig* equals to true:
  - 10.1. ONVIF client invokes **GetAccessPolicy** with user with the user level User credentials (*userLoginUser* and *passwordUser*).
  - 10.2. The DUT responds with **HTTP 401 Unauthorized** error.
11. ONVIF client invokes **CreateUsers** with user with the user level User credentials (*userLoginUser* and *passwordUser*) with parameters
  - *User[0].Username* := "Test"
  - *User[0].Password* := "Test"
  - *User[0].UserLevel* := Administrator
  - Extension skipped
12. The DUT responds with **HTTP 401 Unauthorized** error.
13. ONVIF client invokes **SetSystemDateAndTime** with user with the user level User credentials (*userLoginUser* and *passwordUser*) with parameters
  - *DateTimeType* := NTP
  - *DaylightSavings* := true
  - *TimeZone* skipped
  - *UTCDateTime* skipped
14. The DUT responds with **HTTP 401 Unauthorized** error.
15. ONVIF Client generates creates user with predefined user level (in *userLevelOperator*) and user login (out *userLoginOperator*) and password (out *passwordOperator*) by following the procedure mentioned in Annex 0.
16. ONVIF client invokes **SetScopes** with user with the user level Operator credentials (*userLoginOperator* and *passwordOperator*) with parameters
  - *Scopes[0]* := "onvif://www.onvif.org/location/test"
17. The DUT responds with **HTTP 401 Unauthorized** error.
18. ONVIF client invokes **SetDiscoveryMode** with user with the user level Operator credentials (*userLoginOperator* and *passwordOperator*) with parameters
  - *DiscoveryMode* := Discoverable
19. The DUT responds with **HTTP 401 Unauthorized** error.
20. If *cap.Security* contains *AccessPolicyConfig* and *cap.Security.AccessPolicyConfig* equals to true:
  - 20.1. ONVIF client invokes **GetAccessPolicy** with user with the user level Operator credentials (*userLoginOperator* and *passwordOperator*).



20.2. The DUT responds with **HTTP 401 Unauthorized** error.

21. ONVIF client invokes **CreateUsers** with user with the user level Operator credentials (*userLoginOperator* and *passwordOperator*) with parameters

- User[0].Username := "Test"
- User[0].Password := "Test"
- User[0].UserLevel := Administrator
- Extension skipped

22. The DUT responds with **HTTP 401 Unauthorized** error.

23. ONVIF client invokes **SetSystemDateAndTime** with user with the user level Operator credentials (*userLoginOperator* and *passwordOperator*) with parameters

- DateTimeType := NTP
- DaylightSavings := true
- TimeZone skipped
- UTCDateTime skipped

24. The DUT responds with **HTTP 401 Unauthorized** error.

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

The DUT allowed User access to the **SetScopes** command.

The DUT allowed User access to the **SetDiscoveryMode** command.

The DUT allowed User access to the **GetAccessPolicy** command.

The DUT allowed User access to the **CreateUsers** command.

The DUT allowed User access to the **SetSystemDateAndTime** command.

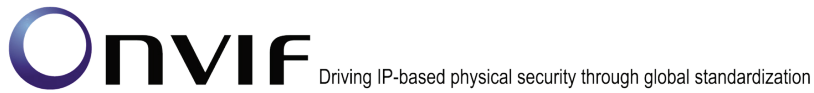
The DUT allowed Operator access to the **SetScopes** command.

The DUT allowed Operator access to the **SetDiscoveryMode** command.

The DUT allowed Operator access to the **GetAccessPolicy** command.

The DUT allowed Operator access to the **CreateUsers** command.

The DUT allowed Operator access to the **SetSystemDateAndTime** command.



### **4.3 Monitoring Events**

#### **4.3.1 Processor Usage event**

**Test Label:** Processor Usage event

**Test Case ID:** QUICK\_INSTALL-3-1-1

**ONVIF Core Specification Coverage:** Monitoring Event ProcessorUsage

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Monitoring/ProcessorUsage event

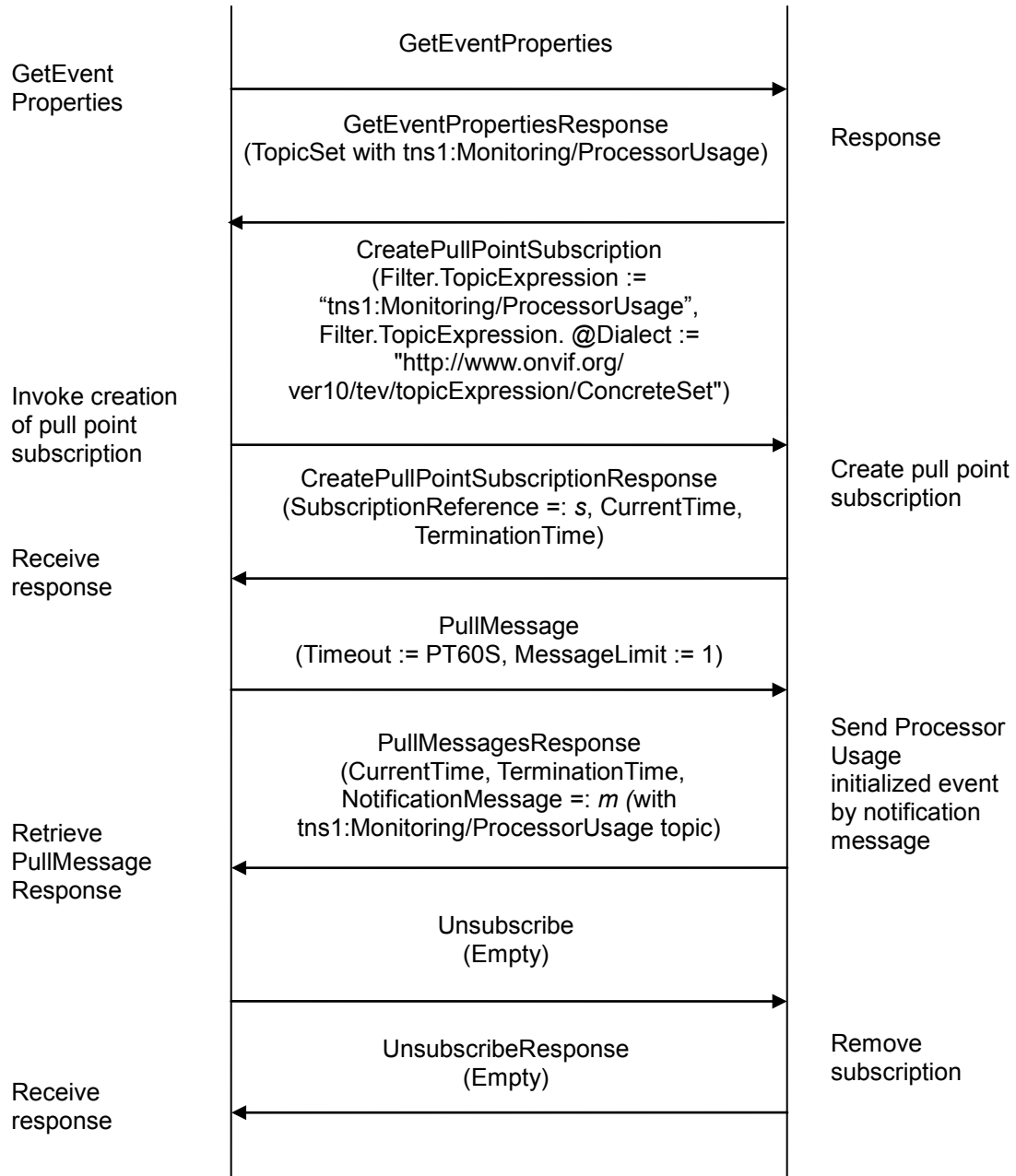
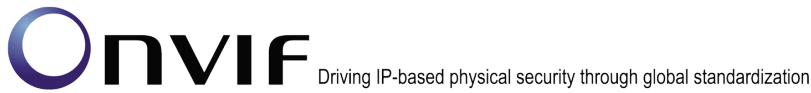
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Monitoring/ProcessorUsage event generation after subscription and to verify tns1:Monitoring/ProcessorUsage event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/ProcessorUsage event is supported by the DUT as indicated by the GetEventPropertiesResponse.

**Test Configuration:** ONVIF Client and DUT

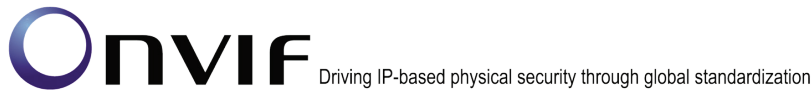
**Test Sequence:**



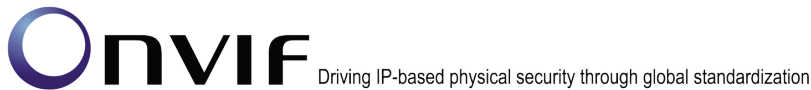
**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters





- TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: *topicSet*
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain tns1:Monitoring/ProcessorUsage topic, FAIL the test and skip other steps.
  6. ONVIF Client verifies tns1:Monitoring/ProcessorUsage topic (*processorUsageTopic*) from *topicSet*:
    - 6.1. If *processorUsageTopic*.MessageDescription.IsProperty is skipped or equals false, FAIL the test and skip other steps.
    - 6.2. If *processorUsageTopic* does not contain MessageDescription.Source.SimpleItemDescription item with Name = "Token", FAIL the test and skip other steps.
    - 6.3. If *processorUsageTopic*.MessageDescription.Source.SimpleItemDescription with Name = "Token" does not have Type = "tt:ReferenceToken", FAIL the test and skip other steps.
    - 6.4. If *processorUsageTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Value", FAIL the test and skip other steps.
    - 6.5. If *processorUsageTopic*.MessageDescription.Data.SimpleItemDescription item with Name = "Value" does not have Type = "xs:float", FAIL the test and skip other steps.
  7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
    - Filter.TopicExpression := "tns1:Monitoring/ProcessorUsage"
    - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
  8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
    - SubscriptionReference =: *s*
    - CurrentTime
    - TerminationTime
  9. Until *timeout1* timeout expires, repeat the following steps:
    - 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
      - Timeout := PT60S
      - MessageLimit := 1
    - 9.2. The DUT responds with **PullMessagesResponse** message with parameters



- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Monitoring/ProcessorUsage, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Source.SimpleItem.Token, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Source.SimpleItem.Token has value type different from tt:ReferenceToken type, FAIL the test and go to the step 10.

9.3.4. If *m* does not contain Message.Message.Data.SimpleItem.Value, FAIL the test and go to the step 10.

9.3.5. If *m*.Message.Message.Data.SimpleItem.Value has value type different from xs:float type, FAIL the test and go to the step 10.

9.3.6. If *m*.Message.Message.Data.SimpleItem.Value value is outside of range 0 to 100, FAIL the test and go to the step 10.

9.3.7. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

#### PASS –

The DUT passed all assertions.

#### FAIL –

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

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

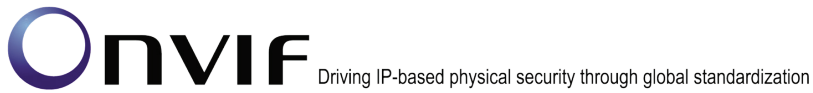
The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

#### 4.3.2 Last Reset event

**Test Label:** Last Reset event



**Test Case ID:** QUICK\_INSTALL-3-1-2

**ONVIF Core Specification Coverage:** Monitoring Event LastReset

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Monitoring/OperatingTime/LastReset event

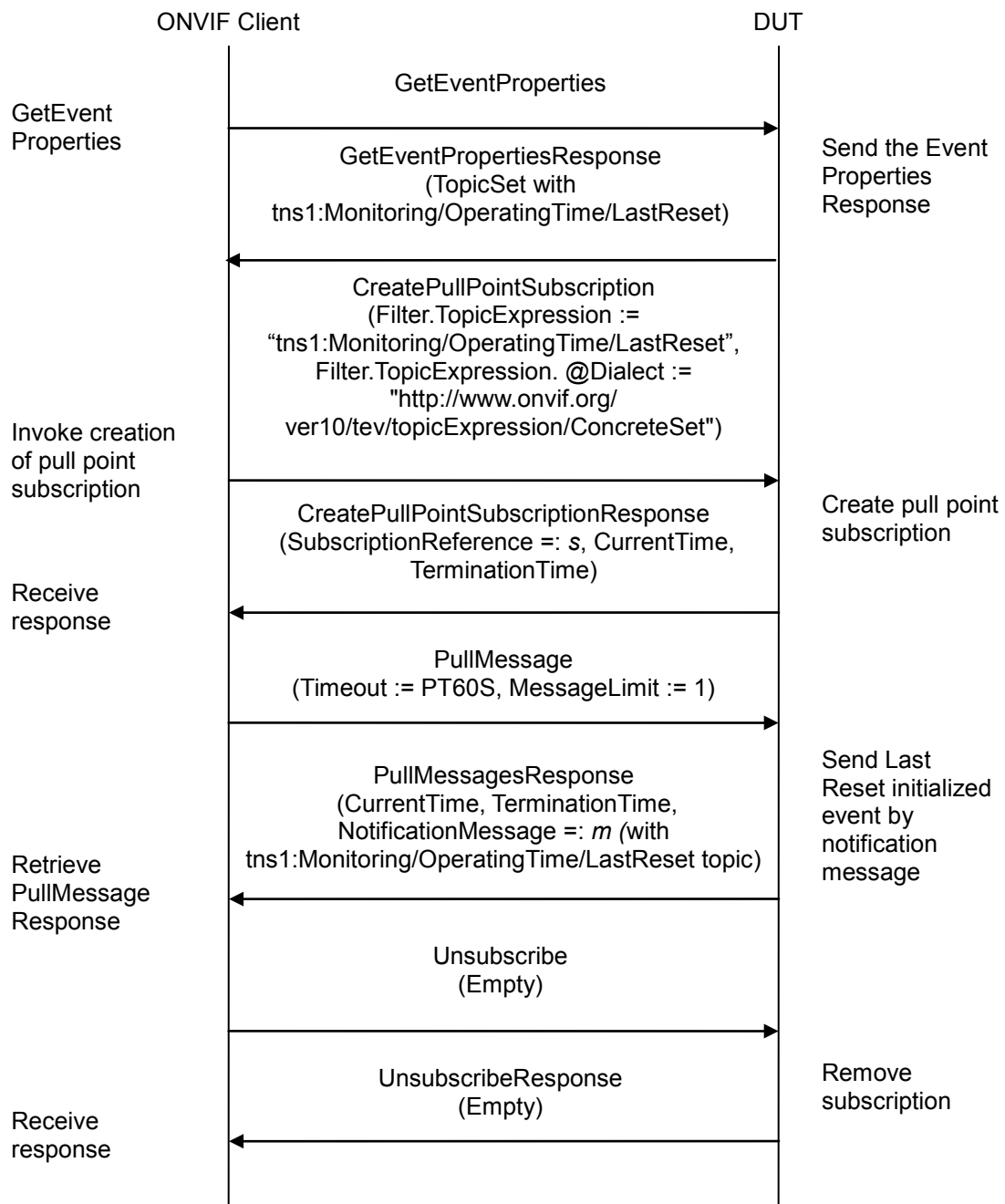
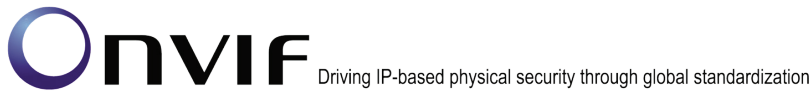
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Monitoring/OperatingTime/LastReset event generation after subscription and to verify tns1:Monitoring/OperatingTime/LastReset event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/OperatingTime/LastReset event is supported by the DUT as indicated by the GetEventPropertiesResponse.

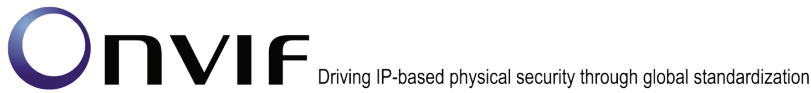
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**

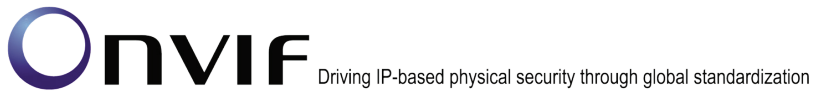


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.



4. The DUT responds with a **GetEventPropertiesResponse** message with parameters
  - TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: *topicSet*
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain tns1:Monitoring/OperatingTime/LastReset topic, FAIL the test and skip other steps.
6. ONVIF Client verifies tns1:Monitoring/OperatingTime/LastReset topic (*lastResetTopic*) from *topicSet*:
  - 6.1. If *lastResetTopic*.MessageDescription.IsProperty is skipped or equals to false, FAIL the test and skip other steps.
  - 6.2. If *lastResetTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Status", FAIL the test and skip other steps.
  - 6.3. If *lastResetTopic*.MessageDescription.Data.SimpleItemDescription with Name = "Status" does not have Type = "xs:dateTime", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - Filter.TopicExpression := "tns1:Monitoring/OperatingTime/LastReset"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - SubscriptionReference =: *s*
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
  - 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
    - Timeout := PT60S
    - MessageLimit := 1
  - 9.2. The DUT responds with **PullMessagesResponse** message with parameters
    - CurrentTime
    - TerminationTime



- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Monitoring/OperatingTime/LastReset, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 10.

9.3.4. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

#### PASS –

The DUT passed all assertions.

#### FAIL –

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

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

#### 4.3.3 Last Reboot event

**Test Label:** Last Reboot event

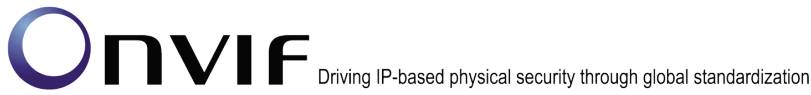
**Test Case ID:** QUICK\_INSTALL-3-1-3

**ONVIF Core Specification Coverage:** Monitoring Event LastReboot

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Monitoring/OperatingTime/LastReboot event

**WSDL Reference:** devicemgmt.wsdl and event.wsdl

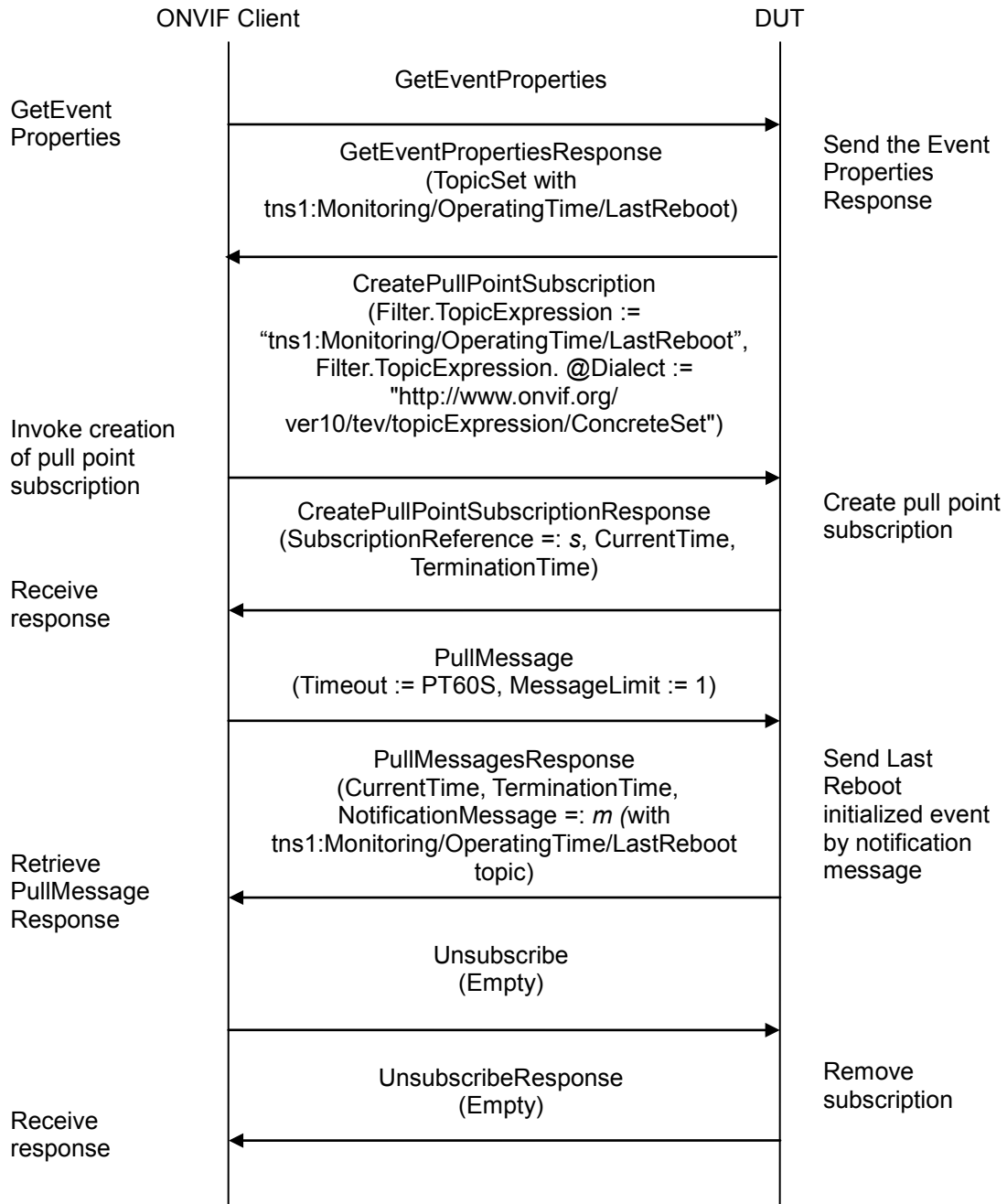
**Test Purpose:** To verify tns1:Monitoring/OperatingTime/LastReboot event generation after subscription and to verify tns1:Monitoring/OperatingTime/LastReboot event format.



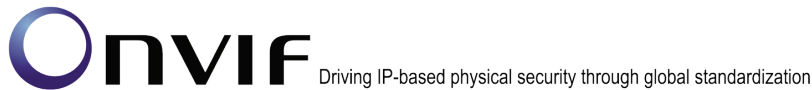
**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/OperatingTime/LastReboot event is supported by the DUT as indicated by the GetEventPropertiesResponse.

**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**

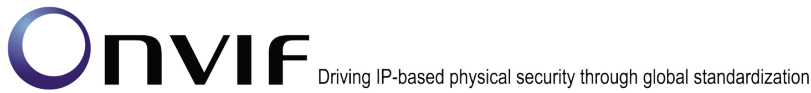


**Test Procedure:**



1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters
  - TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: *topicSet*
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain tns1:Monitoring/OperatingTime/LastReboot topic, FAIL the test and skip other steps.
6. ONVIF Client verifies tns1:Monitoring/OperatingTime/LastReboot topic (*lastRebootTopic*) from *topicSet*:
  - 6.1. If *lastRebootTopic*.MessageDescription.IsProperty is skipped or equals to false, FAIL the test and skip other steps.
  - 6.2. If *lastRebootTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Status", FAIL the test and skip other steps.
  - 6.3. If *lastRebootTopic*.MessageDescription.Data.SimpleItemDescription with Name = "Status" does not have Type = "xs:dateTime", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - Filter.TopicExpression := "tns1:Monitoring/OperatingTime/LastReboot"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - SubscriptionReference =: *s*
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
  - 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
    - Timeout := PT60S
    - MessageLimit := 1





9.2. The DUT responds with **PullMessagesResponse** message with parameters

- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Monitoring/OperatingTime/LastReboot, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 10.

9.3.4. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

#### PASS –

The DUT passed all assertions.

#### FAIL –

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

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

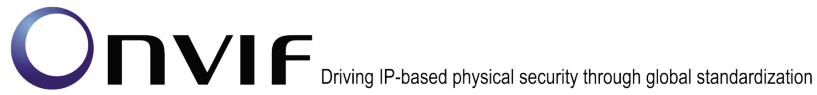
#### 4.3.4 Last Reboot event (Status change)

**Test Label:** Last Reboot event (Status change)

**Test Case ID:** QUICK\_INSTALL-3-1-4

**ONVIF Core Specification Coverage:** Monitoring Event LastReboot

**Command Under Test:** CreatePullPointSubscription, PullMessages, SystemReboot, Monitoring/OperatingTime/LastReboot event



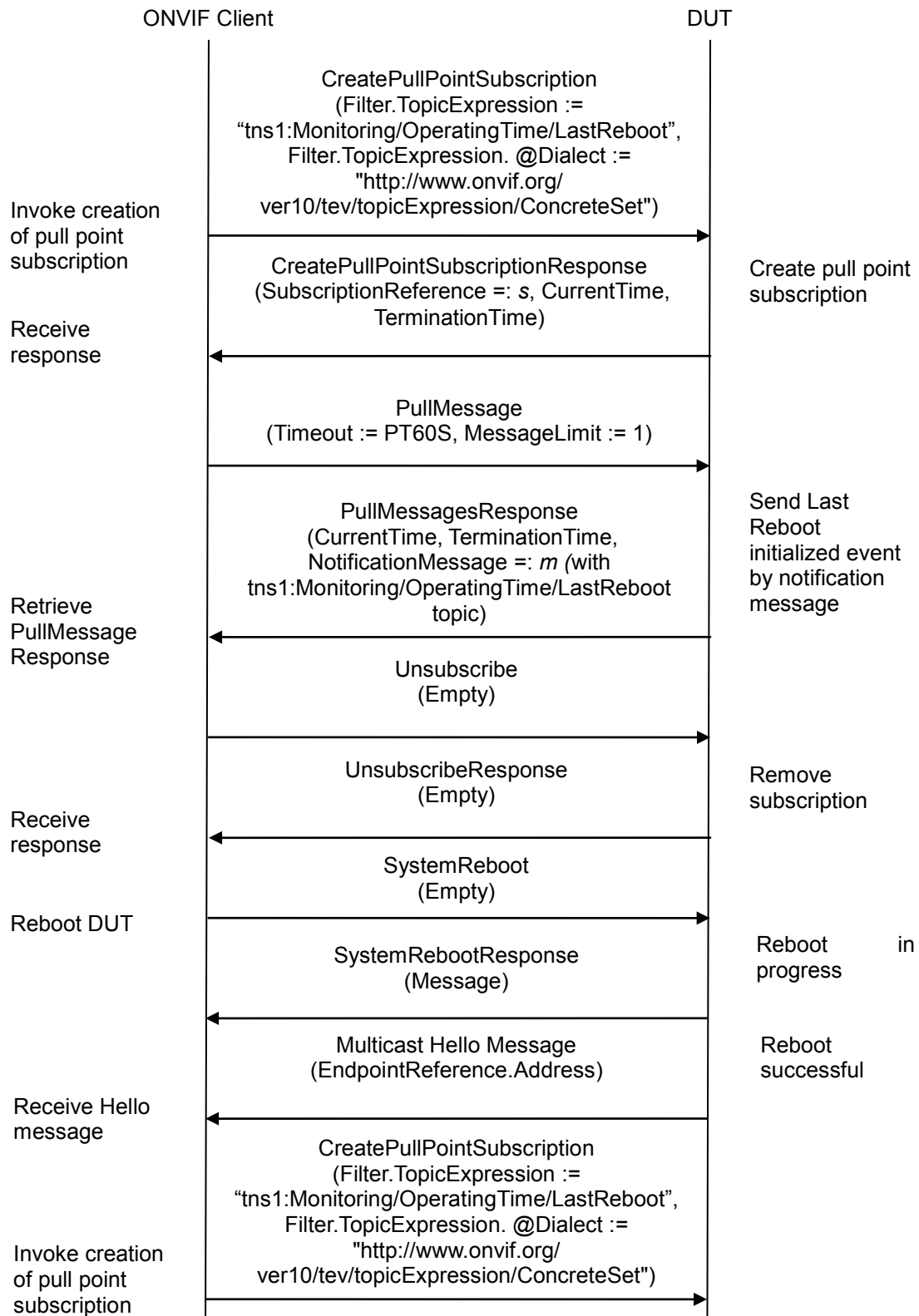
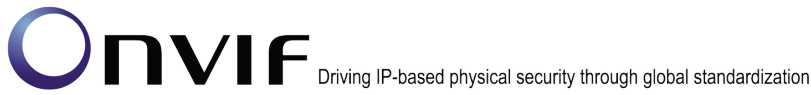
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

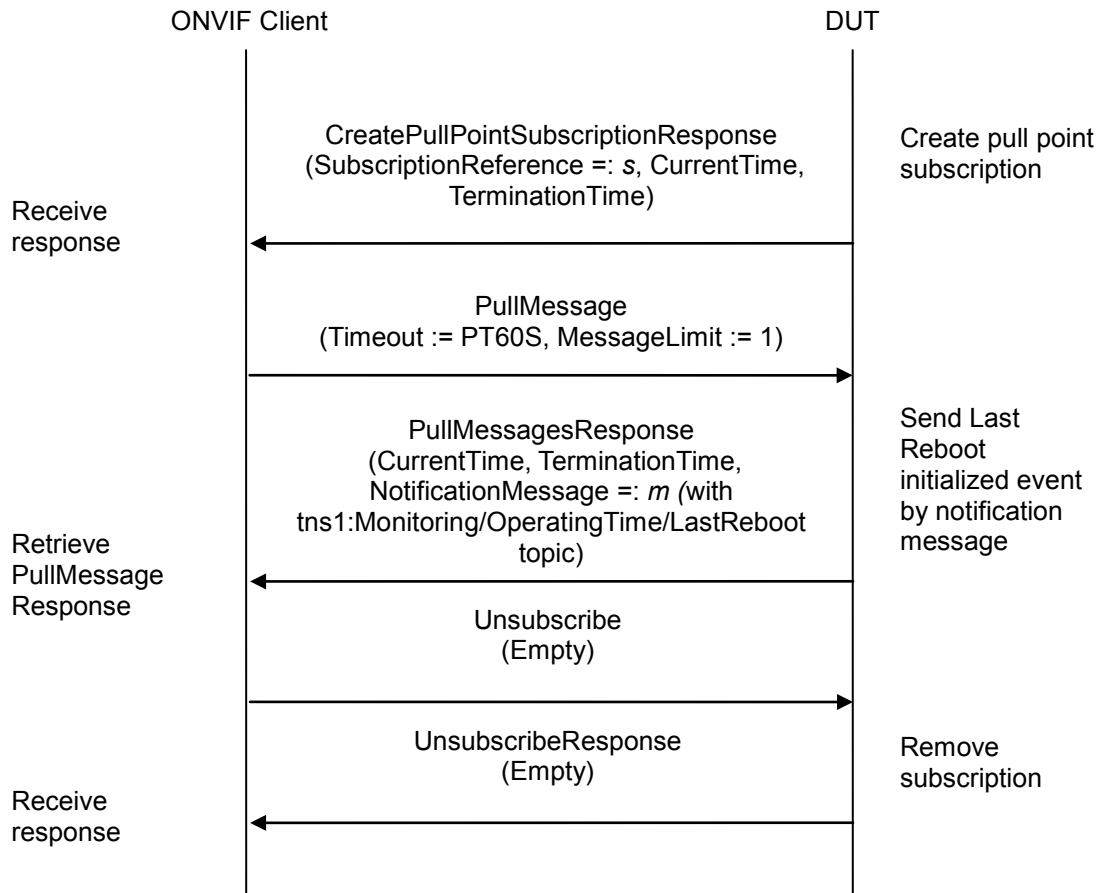
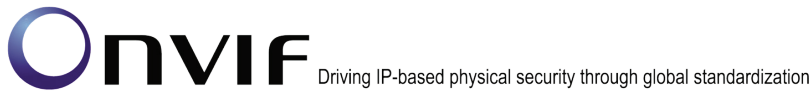
**Test Purpose:** To verify that the last reboot event signals correct values.

**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/OperatingTime/LastReboot event is supported by the DUT as indicated by the GetEventPropertiesResponse. NTP server in network and configured on the DUT.

**Test Configuration:** ONVIF Client and DUT

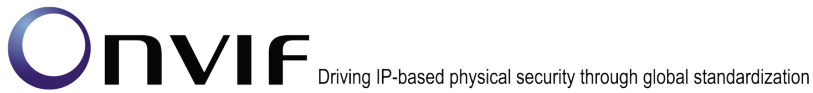
**Test Sequence:**



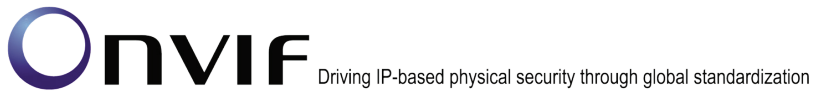


**Test Procedure:**

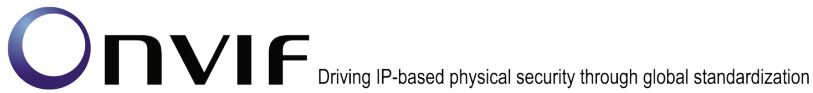
1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - Filter.TopicExpression := “tns1:Monitoring/OperatingTime/LastReboot”
  - Filter.TopicExpression.@Dialect := “http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet”
4. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
5. Until *timeout1* timeout expires, repeat the following steps:



- 5.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
  - Timeout := PT60S
  - MessageLimit := 1
- 5.2. The DUT responds with **PullMessagesResponse** message with parameters
  - CurrentTime
  - TerminationTime
  - NotificationMessage =: *m*
- 5.3. If *m* is not null and *m.Message.Message.PropertyOperation*="Initialized":
  - 5.3.1. If *m* does not contain *Message.Message.Data.SimpleItem.Status*, FAIL the test and go to the step 6.
  - 5.3.2. If *m.Message.Message.Data.SimpleItem.Status* has value type different from *xs:dateTime* type, FAIL the test and go to the step 17.
  - 5.3.3. Set the following:
    - *lastRebootTime* := *m.Message.Message.Data.SimpleItem.Status*
  - 5.3.4. Go to the step 6.
- 5.4. If *timeout1* timeout expires for step 5 without Notification with *PropertyOperation*="Initialized", FAIL the test and go to the step 17.
6. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*. The test should not FAIL even if **Unsubscribe** fails.
7. The DUT responds with **UnsubscribeResponse** message. The test should not FAIL even if **UnsubscribeResponse** message is not received.
8. ONVIF Client invokes **SystemReboot**.
9. The DUT responds with **SystemRebootResponse** message with parameters
  - Message
10. Until *timeout2* timeout expires repeat the following steps:
  - 10.1. The DUT will send Multicast **Hello** message after it is successfully rebooted with parameters:
    - EndpointReference.Address equal to unique endpoint reference of the DUT
    - Types list
    - Scopes list
    - XAddrs list := *xaddrsList*
    - MetadataVersion



- 10.2. If *xaddrsList* contains URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, go to step 12.
11. If *timeout2* timeout expires for step 10 without Hello with URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, FAIL the test and skip other steps.
12. ONVIF client waits for 5 seconds after Hello was received.
13. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - Filter.TopicExpression := “tns1:Monitoring/OperatingTime/LastReboot”
  - Filter.TopicExpression.@Dialect := “http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet”
14. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - SubscriptionReference =: *s*
  - CurrentTime
  - TerminationTime
15. Until *timeout1* timeout expires, repeat the following steps:
  - 15.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
    - Timeout := PT60S
    - MessageLimit := 1
  - 15.2. The DUT responds with **PullMessagesResponse** message with parameters
    - CurrentTime
    - TerminationTime
    - NotificationMessage =: *m*
  - 15.3. If *m* is not null and *m*.Message.Message.PropertyOperation=“Initialized”:
    - 15.3.1. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 16.
    - 15.3.2. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 17.
    - 15.3.3. Set the following:
      - *updatedRebootTime* := *m*.Message.Message.Data.SimpleItem.Status
    - 15.3.4. Go to the step 16.
- 15.4. If *timeout1* timeout expires for step 15 without Notification with PropertyOperation=“Initialized”, FAIL the test and go to the step 17.
16. If *updatedRebootTime* <= *lastRebootTime*, FAIL the test and go to the next step.



17. ONVIF Client sends an **Unsubscribe** to the subscription endpoint s.

18. The DUT responds with **UnsubscribeResponse** message.

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

**Note:** *timeout2* will be taken from Reboot Timeout field of ONVIF Device Test Tool.

**Note:** IPv4 address from Hello shall be used for further test cases. Previous Device service address will be selected if it is present on the XAddr list of Hello message.

#### 4.3.5 Last Clock Synchronization event

**Test Label:** Last Clock Synchronization event

**Test Case ID:** QUICK\_INSTALL-3-1-5

**ONVIF Core Specification Coverage:** Monitoring Event LastClockSynchronization

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Monitoring/OperatingTime/LastClockSynchronization event

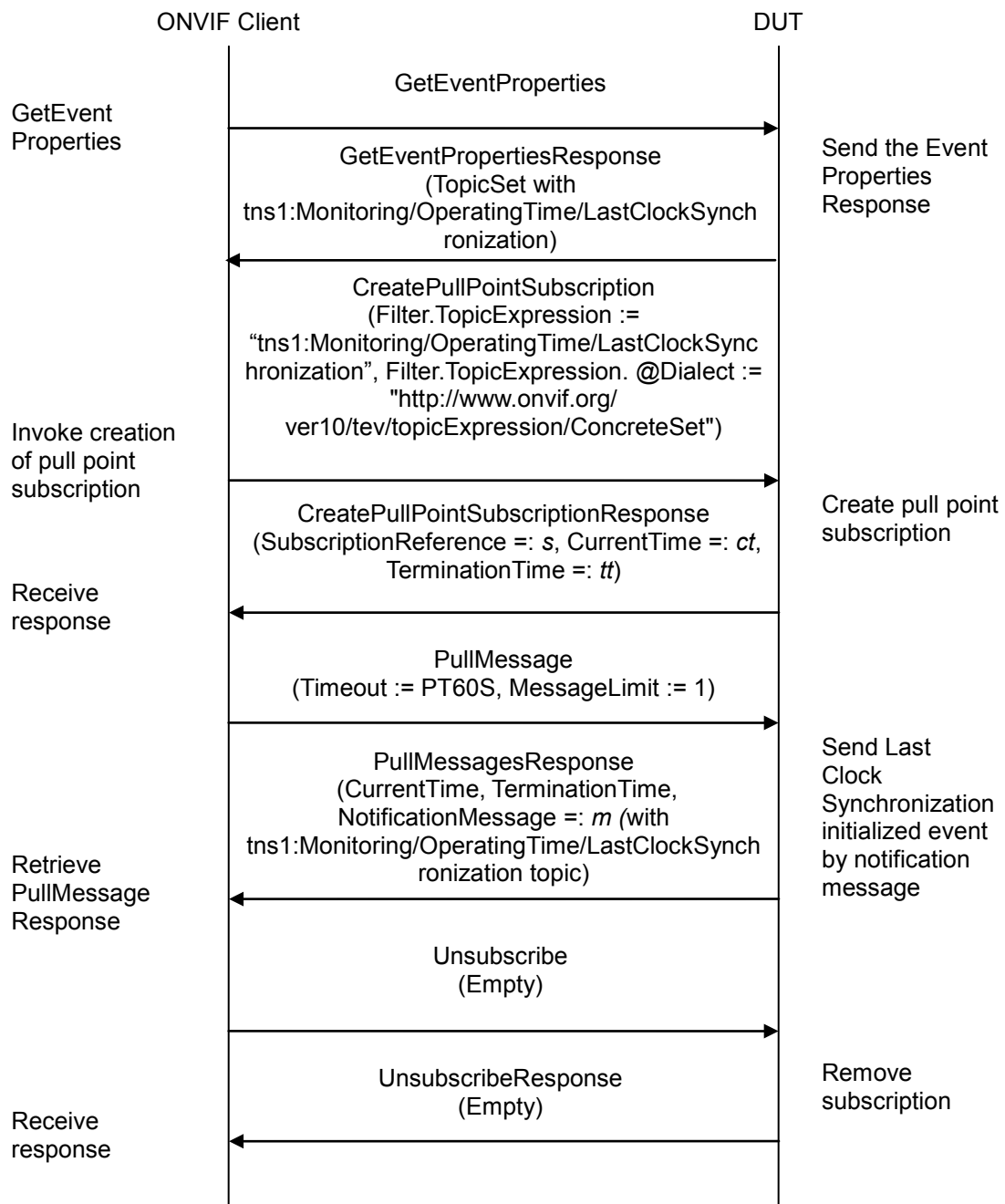
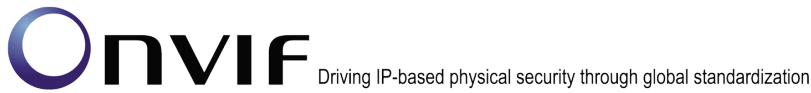
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Monitoring/OperatingTime/LastClockSynchronization event generation after subscription and to verify tns1:Monitoring/OperatingTime/LastClockSynchronization event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/OperatingTime/LastClockSynchronization event is supported by the DUT as indicated by the GetEventPropertiesResponse.

**Test Configuration:** ONVIF Client and DUT

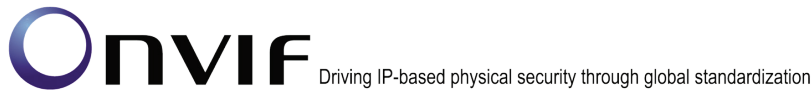
**Test Sequence:**



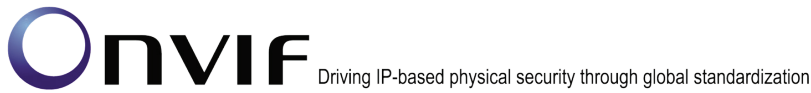
**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters





- TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: *topicSet*
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain `tns1:Monitoring/OperatingTime/LastClockSynchronization` topic, FAIL the test and skip other steps.
  6. ONVIF Client verifies `tns1:Monitoring/OperatingTime/LastClockSynchronization` topic (*lastClockSynchronizationTopic*) from *topicSet*:
    - 6.1. If *lastClockSynchronizationTopic*.MessageDescription.IsProperty is skipped or equals to false, FAIL the test and skip other steps.
    - 6.2. If *lastClockSynchronizationTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Status", FAIL the test and skip other steps.
    - 6.3. If *lastClockSynchronizationTopic*.MessageDescription.Data.SimpleItemDescription with Name = "Status" does not have Type = "xs:dateTime", FAIL the test and skip other steps.
  7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
    - Filter.TopicExpression := "tns1:Monitoring/OperatingTime/LastClockSynchronization"
    - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
  8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
    - SubscriptionReference =: *s*
    - CurrentTime
    - TerminationTime
  9. Until *timeout1* timeout expires, repeat the following steps:
    - 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
      - Timeout := PT60S
      - MessageLimit := 1
    - 9.2. The DUT responds with **PullMessagesResponse** message with parameters
      - CurrentTime
      - TerminationTime



- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Monitoring/OperatingTime/LastClockSynchronization, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 10.

9.3.4. If *m*.Message.Message.Data.SimpleItem.Status has value time in format different from utc format with including the 'Z' indicator, FAIL the test and go to the step 10.

9.3.5. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

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

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

**4.3.6 Last Clock Synchronization change event (SetSystemDateAndTime)**

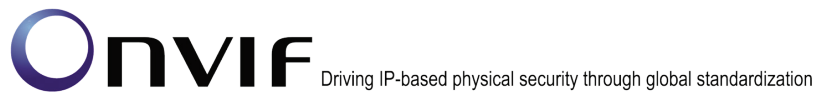
**Test Label:** Last Clock Synchronization change event (SetSystemDateAndTime)

**Test Case ID:** QUICK\_INSTALL-3-1-6

**ONVIF Core Specification Coverage:** Monitoring Event LastClockSynchronization

**Command Under Test:** GetServices, CreatePullPointSubscription, PullMessages, SetSystemDateAndTime, Monitoring/OperatingTime/LastClockSynchronization event

**WSDL Reference:** devicemgmt.wsdl and event.wsdl

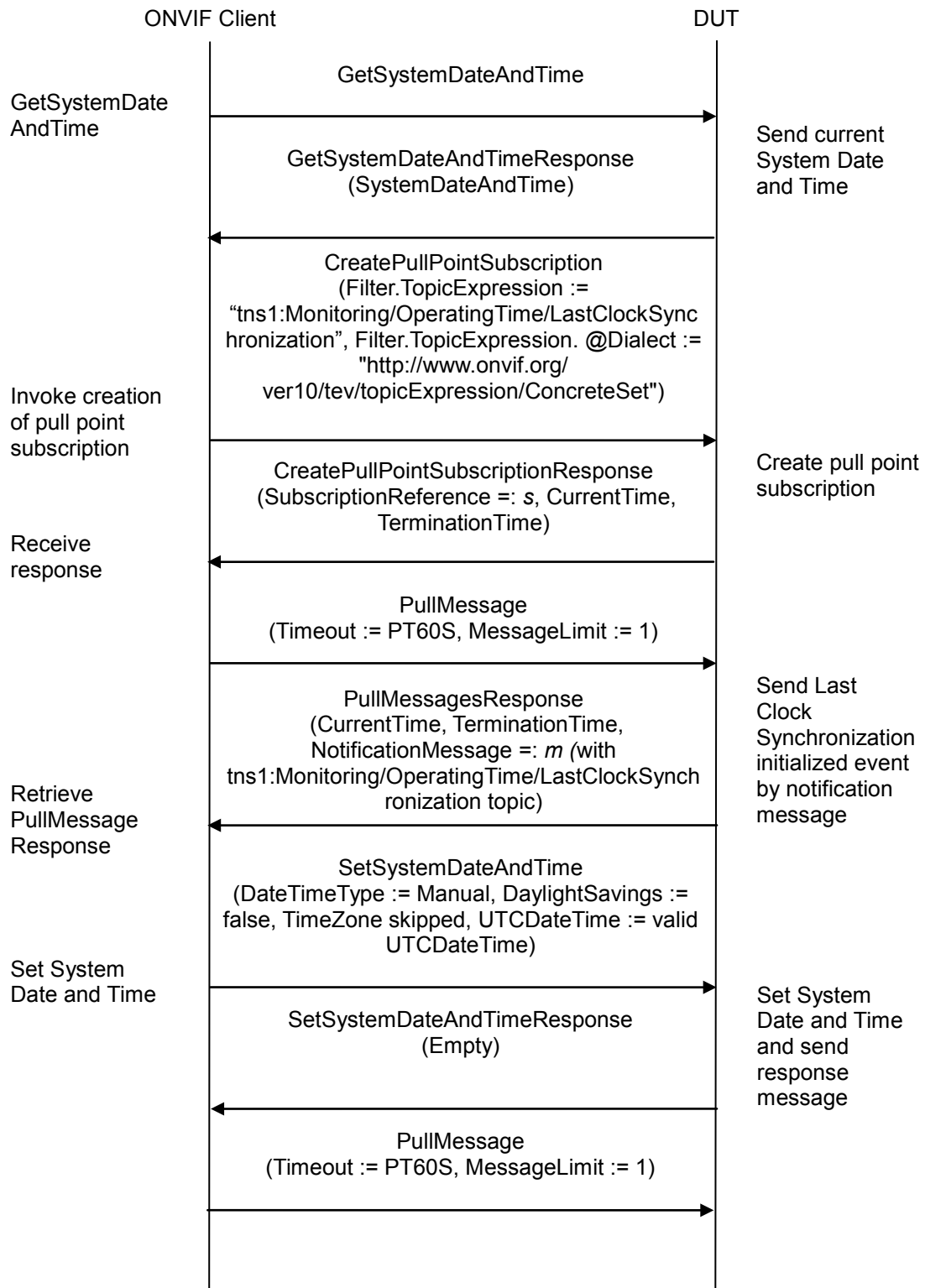
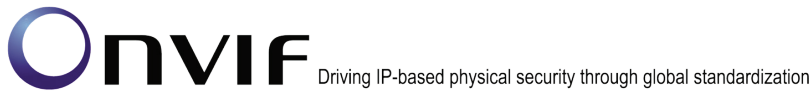


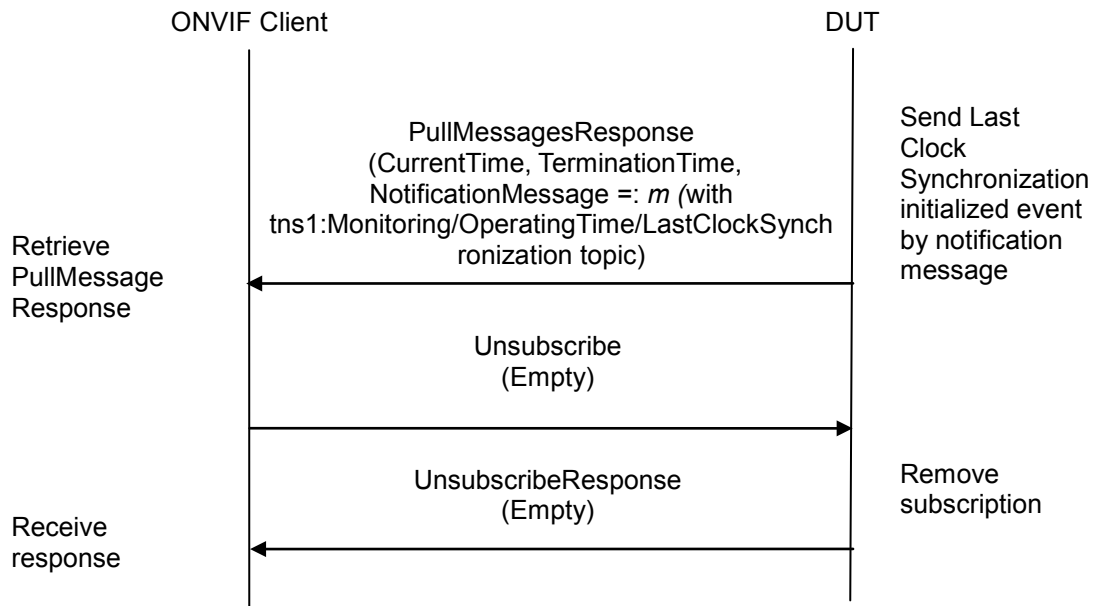
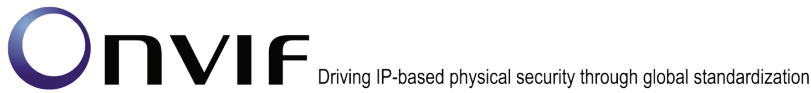
**Test Purpose:** To verify tns1:Monitoring/OperatingTime/LastClockSynchronization event generation after property was changed via SetSystemDateAndTime call and to verify tns1:Monitoring/OperatingTime/LastClockSynchronization event format. To verify that the clock synchronization event signals correct values.

**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/OperatingTime/LastClockSynchronization event is supported by the DUT as indicated by the GetEventPropertiesResponse.

**Test Configuration:** ONVIF Client and DUT

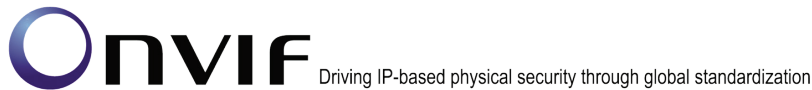
**Test Sequence:**





#### Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetSystemDateAndTime**.
4. The DUT responds with a **GetSystemDateAndTimeResponse** message with parameters
  - *SystemDateAndTime =: initialSystemDateAndTime*
5. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - *Filter.TopicExpression := "tns1:Monitoring/OperatingTime/LastClockSynchronization"*
  - *Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"*
6. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - *SubscriptionReference =: s*
  - *CurrentTime*
  - *TerminationTime*
7. Until *timeout1* timeout expires, repeat the following steps:
  - 7.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
    - *Timeout := PT60S*



- MessageLimit := 1

7.2. The DUT responds with **PullMessagesResponse** message with parameters

- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

7.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

7.3.1. If *m*.TopicDialect does not equal to tns1:Monitoring/OperatingTime/LastClockSynchronization, FAIL the test and go to the step 14.

7.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 14.

7.3.3. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 14.

7.3.4. Go to the step 8.

7.4. If *timeout1* timeout expires for step 7 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 14.

8. Set the following:

- *initialClockSynchronization* := *m*.Message.Message.Data.SimpleItem.Status

9. ONVIF Client waits for 1 second.

10. ONVIF Client invokes **SetSystemDateAndTime** with parameters

- DateTimeType := Manual
- DaylightSavings := false
- TimeZone skipped
- UTCDateTime := *initialSystemDateAndTime.UTCDateTime* received from DUT on step 4. If *initialSystemDateAndTime.UTCDateTime* is empty, then system date and time should be used to populate this parameter.

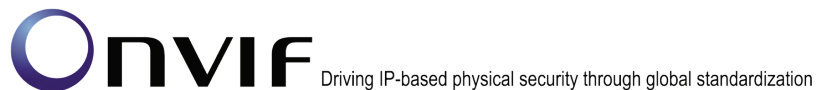
11. The DUT responds with a **SetSystemDateAndTimeResponse** message.

12. Until *timeout1* timeout expires, repeat the following steps:

12.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters

- Timeout := PT60S
- MessageLimit := 1

12.2. The DUT responds with **PullMessagesResponse** message with parameters



- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

12.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Changed" ONVIF Client verifies *m*:

12.3.1. If *m*.TopicDialect does not equal to tns1:Monitoring/OperatingTime/LastClockSynchronization, FAIL the test and go to the step 14.

12.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 14.

12.3.3. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 14.

12.3.4. Set the following:

- *lastClockSynchronization* := *m*.Message.Message.Data.SimpleItem.Status

12.3.5. Go to the step 13.

12.4. If *timeout1* timeout expires for step 11 without Notification with PropertyOperation="Changed", FAIL the test and go to the step 14.

13. If *lastClockSynchronization* is less or equal to *initialClockSynchronization*, FAIL the test and go to the next step.

14. ONVIF Client restores Default System Date and Time by following the procedure mentioned in Annex A.2.

15. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

16. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

##### PASS –

The DUT passed all assertions.

##### FAIL –

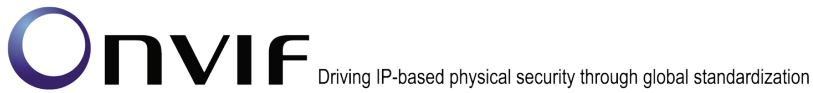
The DUT did not send **GetSystemDateAndTimeResponse** message.

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.



#### **4.3.7 Last Clock Synchronization change event (NTP message)**

**Test Label:** Last Clock Synchronization change event (NTP message)

**Test Case ID:** QUICK\_INSTALL-3-1-7

**ONVIF Core Specification Coverage:** Monitoring Event LastClockSynchronization

**Command Under Test:** GetServices, CreatePullPointSubscription, PullMessages, SetSystemDateAndTime, SetNTP, Monitoring/OperatingTime/LastClockSynchronization event

**WSDL Reference:** devicemgmt.wsdl and event.wsdl

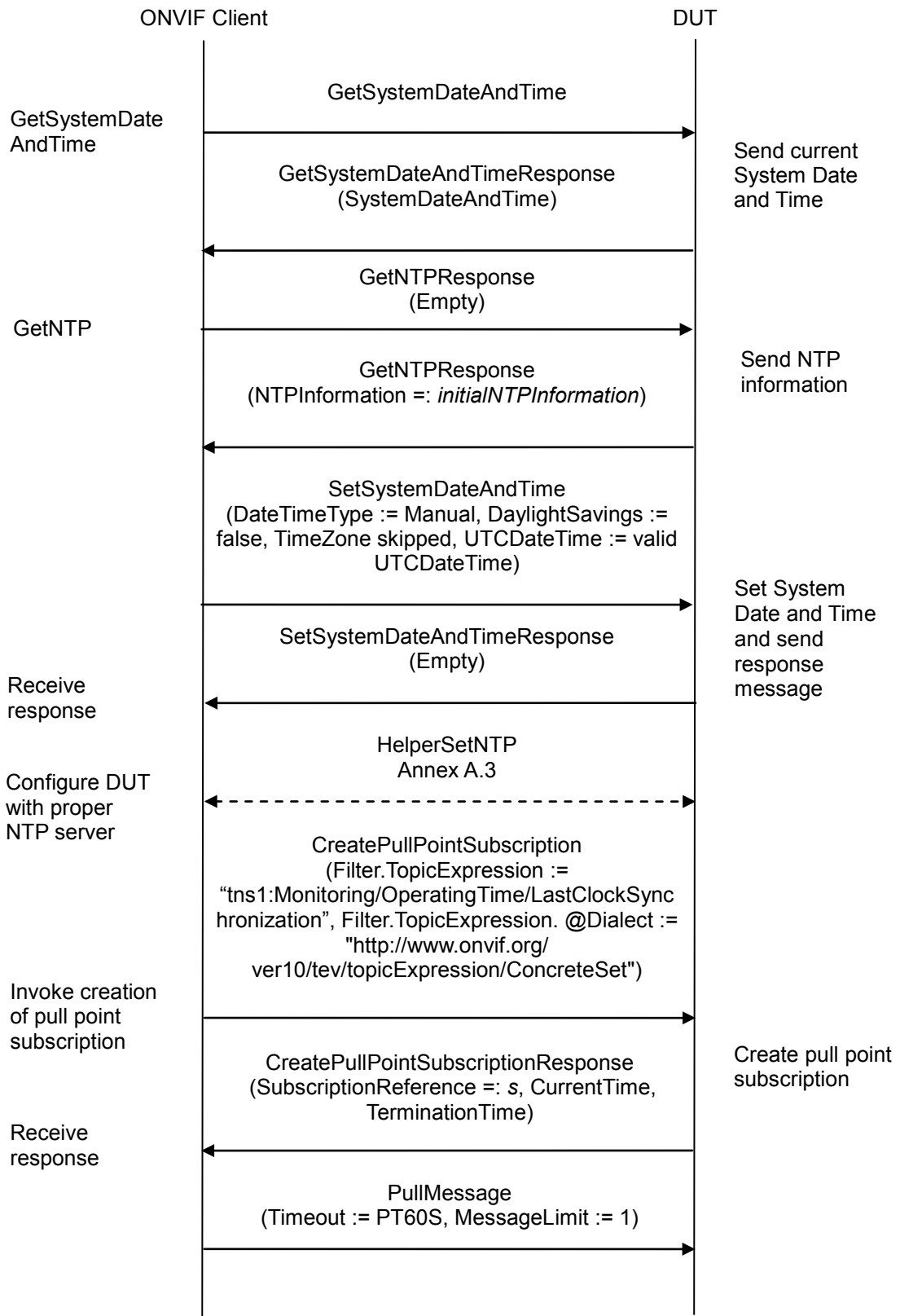
**Test Purpose:** To verify tns1:Monitoring/OperatingTime/LastClockSynchronization event generation after property was changed via an NTP message and to verify tns1:Monitoring/OperatingTime/LastClockSynchronization event format. To verify that the clock synchronization event signals correct values.

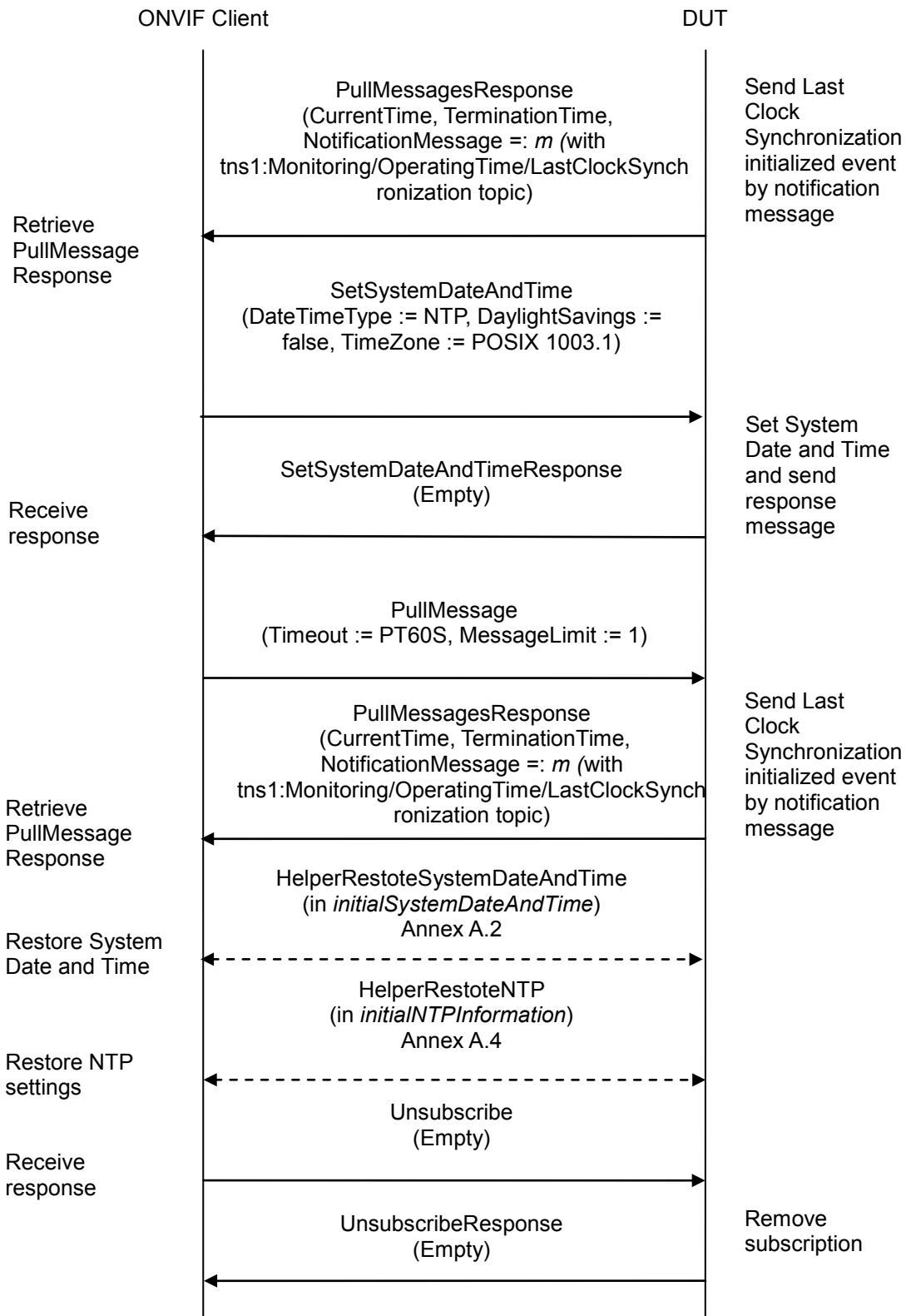
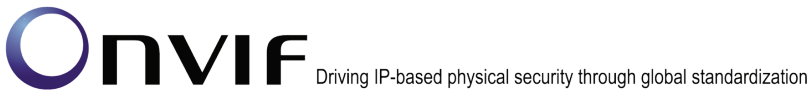
**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/OperatingTime/LastClockSynchronization event is supported by the DUT as indicated by the GetEventPropertiesResponse. NTP is supported by the DUT as indicated by the Network.NTP capability. A valid NTP server address should be configured in the DUT.

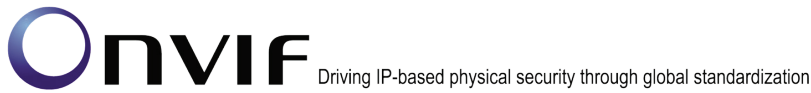
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**



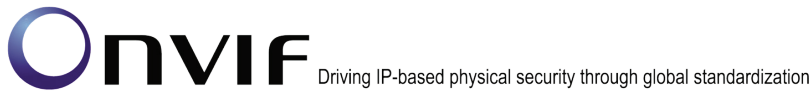






**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetSystemDateAndTime**.
4. The DUT responds with a **GetSystemDateAndTimeResponse** message with parameters
  - SystemDateAndTime =: *initialSystemDateAndTime*
5. ONVIF Client invokes **GetNTP**.
6. The DUT responds with a **GetNTPResponse** message with parameters
  - NTPInformation =: *initialNTPInformation*
7. If *initialSystemDateAndTime.DataTimeType* = NTP, ONVIF Client invokes **SetSystemDateAndTime** with parameters
  - DateTimeType := Manual
  - DaylightSavings := false
  - TimeZone skipped
  - UTCDateTime := *initialSystemDateAndTime.UTCDateTime* received from DUT on step 4. If *initialSystemDateAndTime.UTCDateTime* is empty, then system date and time should be used to populate this parameter.
8. The DUT responds with a **SetSystemDateAndTimeResponse** message.
9. ONVIF Client configures DUT with proper NTP server by following the procedure mentioned in Annex A.3.
10. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - Filter.TopicExpression := "tns1:Monitoring/OperatingTime/LastClockSynchronization"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
11. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
12. Until *timeout1* timeout expires, repeat the following steps:
  - 12.1. ONVIF Client invokes **PullMessages** to the subscription endpoint s with parameters
    - Timeout := PT60S
    - MessageLimit := 1



12.2. The DUT responds with **PullMessagesResponse** message with parameters

- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

12.3. If *m* is not null and *m.Message.Message.PropertyOperation*="Initialized" ONVIF Client verifies *m*:

12.3.1. If *m* does not contain *Message.Message.Data.SimpleItem.Status*, FAIL the test and go to the step 18.

12.3.2. If *m.Message.Message.Data.SimpleItem.Status* has value type different from *xs:dateTime* type, FAIL the test and go to the step 18.

12.3.3. Go to the step 13.

12.4. If *timeout1* timeout expires for step 12 without Notification with *PropertyOperation*="Initialized", FAIL the test and go to the step 18.

13. Set the following:

- *initialClockSynchronization* := *m.Message.Message.Data.SimpleItem.Status*

14. ONVIF Client waits for 1 second.

15. ONVIF Client invokes **SetSystemDateAndTime** with parameters

- DateTimeType := NTP
- DaylightSavings := false
- TimeZone := POSIX 1003.1
- UTCDateTime skipped

16. The DUT responds with a **SetSystemDateAndTimeResponse** message.

17. Set the following:

- *timeOfResponse* := time of SetSystemDateAndTimeResponse message receiving

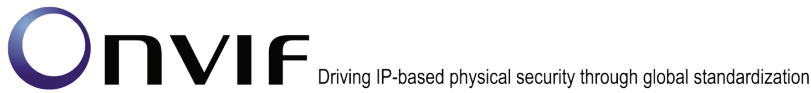
18. Until *timeout1* timeout expires, repeat the following steps:

18.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters

- Timeout := PT60S
- MessageLimit := 1

18.2. The DUT responds with **PullMessagesResponse** message with parameters

- CurrentTime
- TerminationTime



- NotificationMessage =: *m*

18.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Changed" ONVIF Client verifies *m*:

18.3.1. If *m*.TopicDialect does not equal to tns1:Monitoring/OperatingTime/LastClockSynchronization, FAIL the test and go to the step 20.

18.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Status, FAIL the test and go to the step 20.

18.3.3. If *m*.Message.Message.Data.SimpleItem.Status has value type different from xs:dateTime type, FAIL the test and go to the step 20.

18.3.4. Set the following:

- *lastClockSynchronization* := *m*.Message.Message.Data.SimpleItem.Status

18.3.5. Go to the step 19.

18.4. If *timeout1* timeout expires for step 10 without Notification with PropertyOperation="Changed", FAIL the test and go to the step 20.

19. If *lastClockSynchronization* is less or equal to *initialClockSynchronization*, FAIL the test and go to the next step.

20. ONVIF Client restores Default System Date and Time by following the procedure mentioned in Annex A.2.

21. ONVIF Client restores Default NTP settings by following the procedure mentioned in Annex A.4.

22. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

23. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

#### PASS –

The DUT passed all assertions.

#### FAIL –

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

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

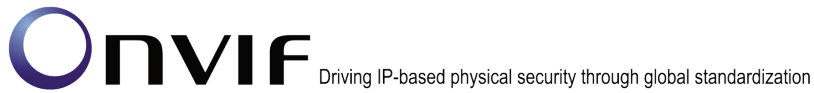
The DUT did not send **CreatePullPointSubscriptionResponse** message.

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.



#### **4.3.8 Last Backup event**

**Test Label:** Last Backup event

**Test Case ID:** QUICK\_INSTALL-3-1-8

**ONVIF Core Specification Coverage:** Monitoring Event Last Backup

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Monitoring/Backup/Last event

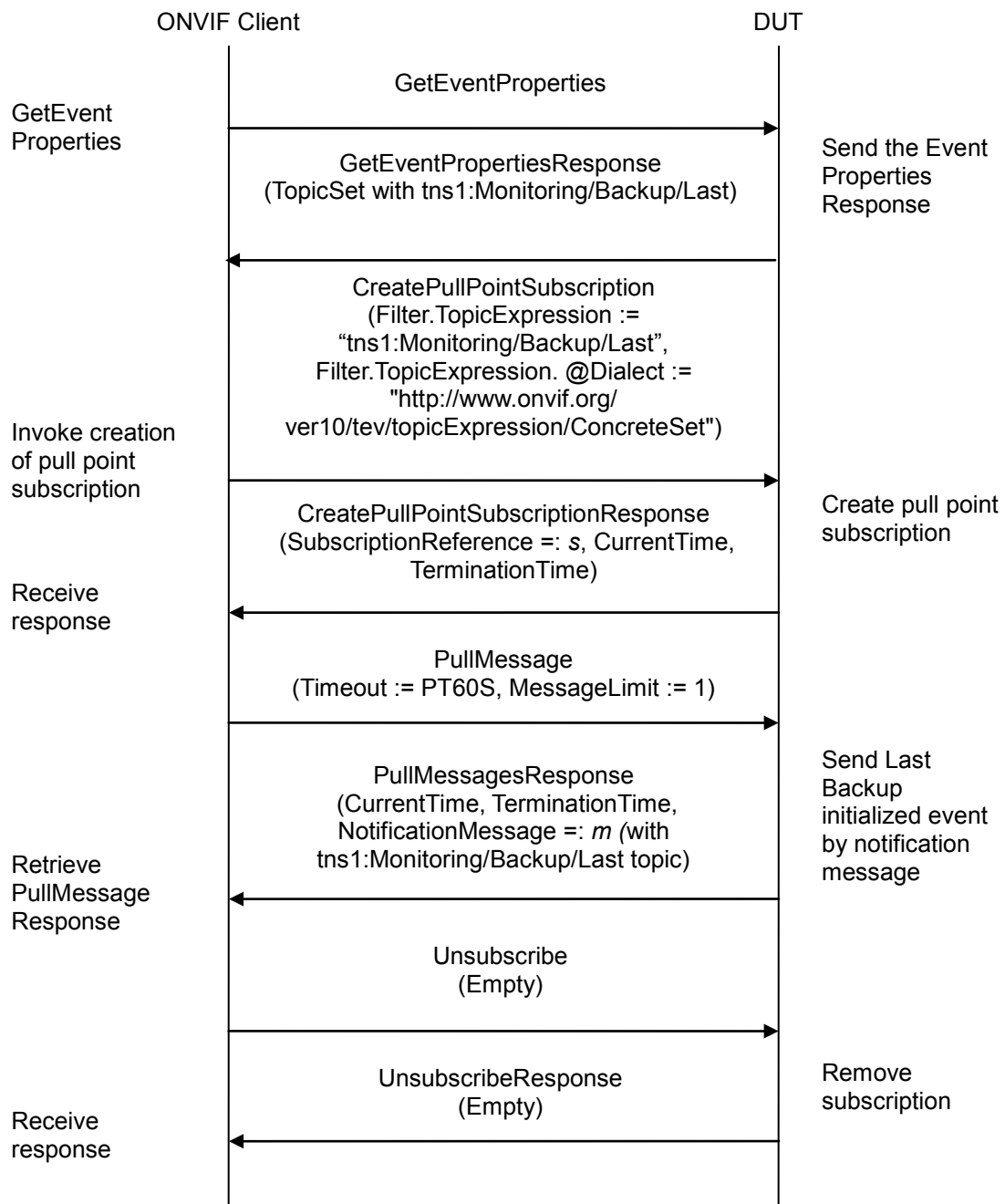
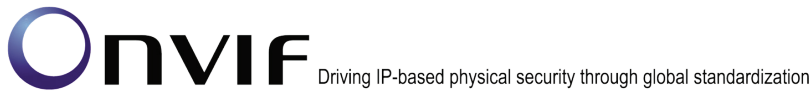
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Monitoring/Backup/Last initial event generation and to verify tns1:Monitoring/Backup/Last event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Monitoring/Backup/Last event is supported by the DUT as indicated by the GetEventPropertiesResponse.

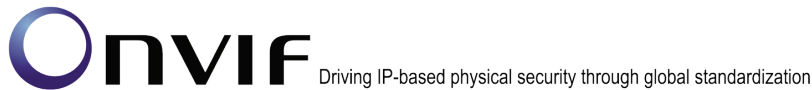
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**



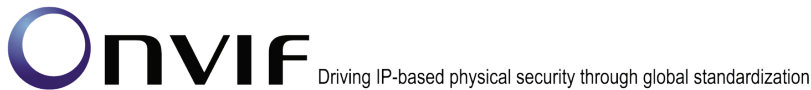
**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters



- TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: topicSet
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain tns1:Monitoring/Backup/Last topic, FAIL the test and skip other steps.
6. ONVIF Client verifies tns1:Monitoring/Backup/Last topic (*lastBackupTopic*) from *topicSet*:
- 6.1. If *lastBackupTopic*.MessageDescription.IsProperty is skipped or equals to false, FAIL the test and skip other steps.
- 6.2. If *lastBackupTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Status", FAIL the test and skip other steps.
- 6.3. If *lastBackupTopic*.MessageDescription.Data.SimpleItemDescription item with Name = "Status" does not have Type = "xs:dateTime", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
- Filter.TopicExpression := "tns1:Monitoring/Backup/Last"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
- SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
- 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
- Timeout := PT60S
  - MessageLimit := 1
- 9.2. The DUT responds with **PullMessagesResponse** message with parameters
- CurrentTime
  - TerminationTime
  - NotificationMessage =: *m*
- 9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client





verifies *m*:

9.3.1. If *m.Topic* does not equal to `tns1:Monitoring/Backup/Last`, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain `Message.Message.Data.SimpleItem.Status`, FAIL the test and go to the step 10.

9.3.3. If *m.Message.Message.Data.SimpleItem.Status* has value type different from `xs:dateTime` type, FAIL the test and go to the step 10.

9.3.4. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with `PropertyOperation="Initialized"`, FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

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

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

#### 4.3.9 Fan Failure event

**Test Label:** Fan Failure event

**Test Case ID:** QUICK\_INSTALL-3-1-9

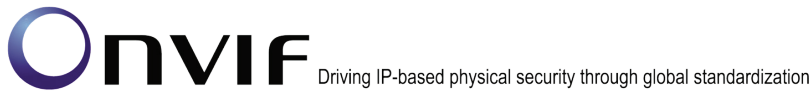
**ONVIF Core Specification Coverage:** Monitoring Event Fan Failure

**Command Under Test:** `GetServices`, `GetEventProperties`, `CreatePullPointSubscription`, `PullMessages`, `Device/HardwareFailure/FanFailure` event format

**WSDL Reference:** `devicemgmt.wsdl` and `event.wsdl`

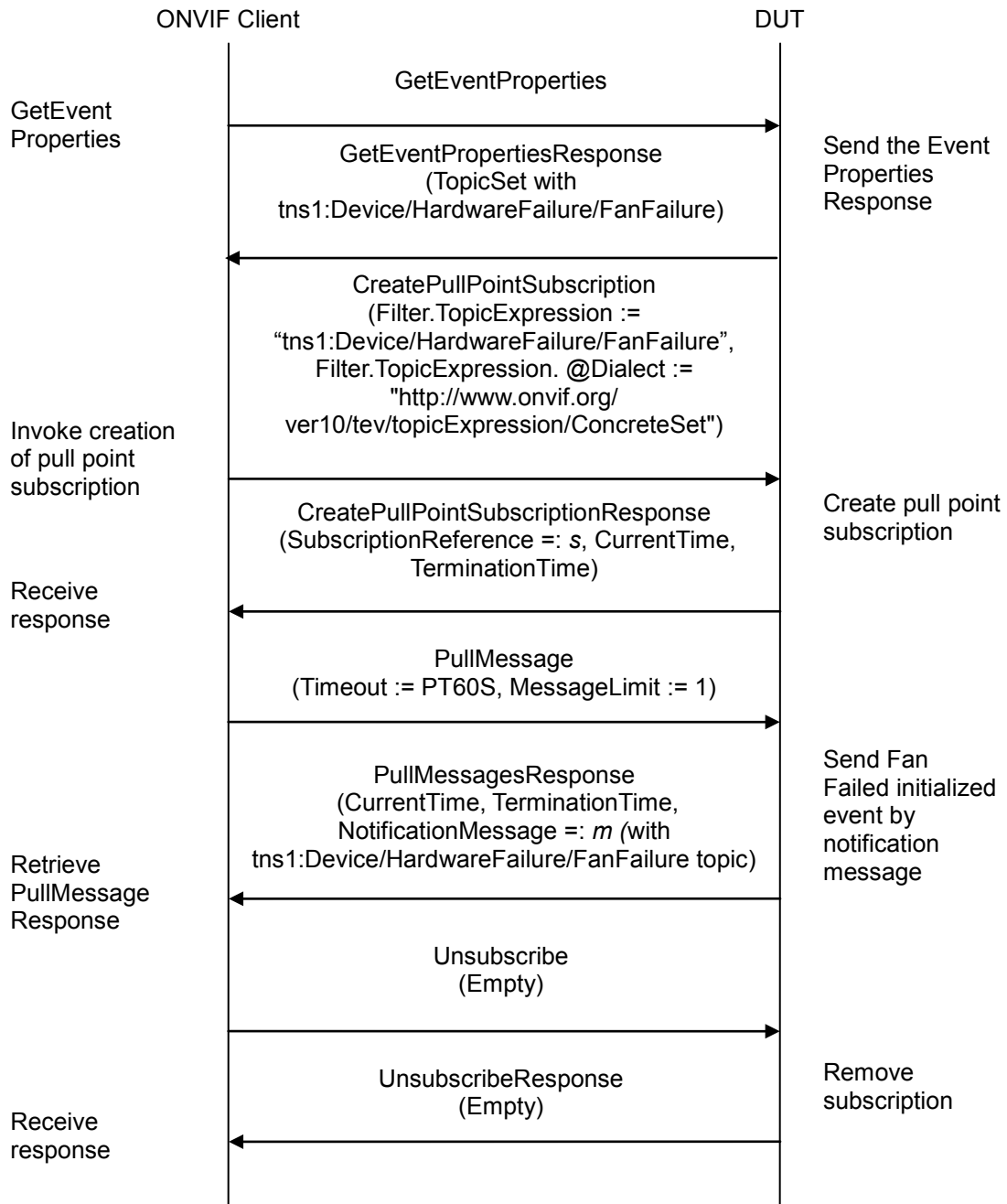
**Test Purpose:** To verify `tns1:Device/HardwareFailure/FanFailure` event generation and to verify `tns1:Device/HardwareFailure/FanFailure` event format.

**Pre-requisite:** Event Service was received from the DUT. `tns1:Device/HardwareFailure/FanFailure` event is supported by the DUT as indicated by the `GetEventPropertiesResponse`.



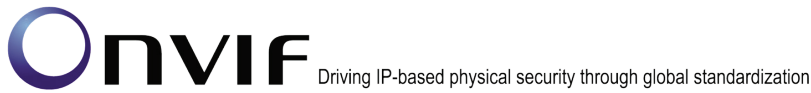
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**

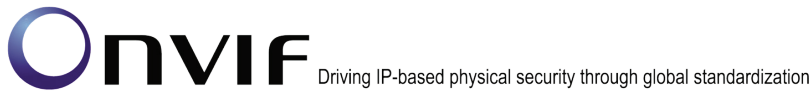


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.



3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters
  - TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: topicSet
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain tns1:Device/HardwareFailure/FanFailure topic, FAIL the test and skip other steps.
6. ONVIF Client verifies tns1:Device/HardwareFailure/FanFailure topic (*fanFailedTopic*) from *topicSet*:
  - 6.1. If *fanFailureTopic.MessageDescription.IsProperty* is skipped or equals to false, FAIL the test and skip other steps.
  - 6.2. If *fanFailureTopic* does not contain *MessageDescription.Source.SimpleItemDescription* item with Name = "Token", FAIL the test and skip other steps.
  - 6.3. If *fanFailureTopic.MessageDescription.Source.SimpleItemDescription* item with Name = "Token" does not have Type = "tt:ReferenceToken", FAIL the test and skip other steps.
  - 6.4. If *fanFailureTopic* does not contain *MessageDescription.Data.SimpleItemDescription* item with Name = "Failed", FAIL the test and skip other steps.
  - 6.5. If *fanFailureTopic.MessageDescription.Data.SimpleItemDescription* item with Name = "Failed" does not have Type = "xs:boolean", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
  - Filter.TopicExpression := "tns1:Device/HardwareFailure/FanFailure"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
  - SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
  - 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
    - Timeout := PT60S



- MessageLimit := 1

9.2. The DUT responds with **PullMessagesResponse** message with parameters

- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Device/HardwareFailure/FanFailure, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Source.SimpleItem.Token, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Source.SimpleItem.Token has value type different from tt:ReferenceToken type, FAIL the test and go to the step 10.

9.3.4. If *m* does not contain Message.Message.Data.SimpleItem.Failed, FAIL the test and go to the step 10.

9.3.5. If *m*.Message.Message.Data.SimpleItem.Failed has value type different from xs:boolean type, FAIL the test and go to the step 10.

9.3.6. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

##### PASS –

The DUT passed all assertions.

##### FAIL –

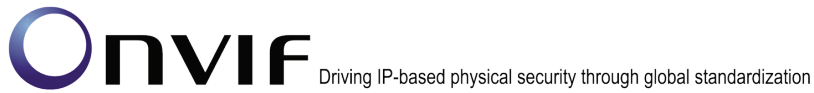
The DUT did not send **GetEventPropertiesResponse** message.

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.



#### **4.3.10 Power Supply Failure event**

**Test Label:** Power Supply Failure event

**Test Case ID:** QUICK\_INSTALL-3-1-10

**ONVIF Core Specification Coverage:** Monitoring Event Power Supply Failure

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Device/HardwareFailure/PowerSupplyFailure event

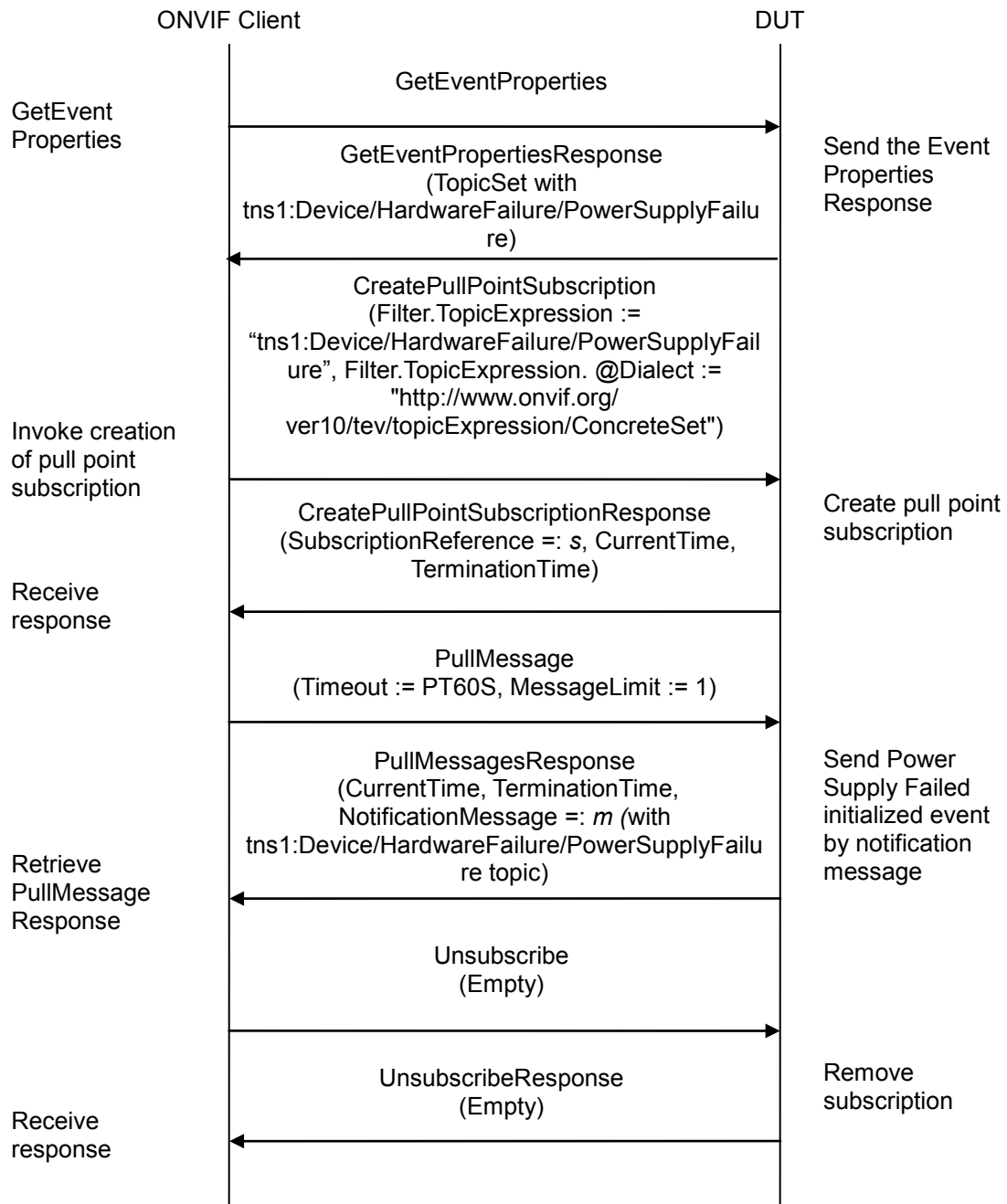
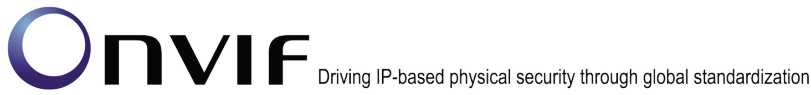
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Device/HardwareFailure/PowerSupplyFailure event generation and to verify tns1:Device/HardwareFailure/PowerSupplyFailure event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Device/HardwareFailure/PowerSupplyFailure event is supported by the DUT as indicated by the GetEventPropertiesResponse.

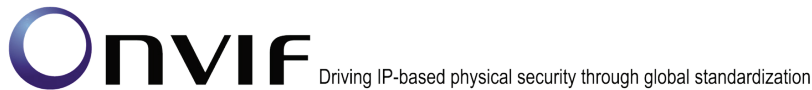
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**

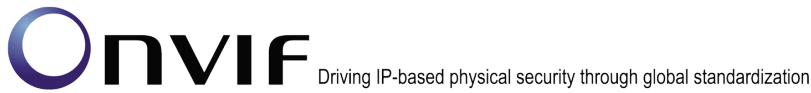


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters



- TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: topicSet
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain `tns1:Device/HardwareFailure/PowerSupplyFailure` topic, FAIL the test and skip other steps.
6. ONVIF Client verifies `tns1:Device/HardwareFailure/PowerSupplyFailure` topic (*powerSupplyFailureTopic*) from *topicSet*:
- 6.1. If *powerSupplyFailureTopic.MessageDescription.IsProperty* is skipped or equals to false, FAIL the test and skip other steps.
- 6.2. If *powerSupplyFailureTopic* does not contain `MessageDescription.Source.SimpleItemDescription` item with Name = "Token", FAIL the test and skip other steps.
- 6.3. If *powerSupplyFailureTopic.MessageDescription.Source.SimpleItemDescription* item with Name = "Token" does not have Type = "tt:ReferenceToken", FAIL the test and skip other steps.
- 6.4. If *powerSupplyFailedTopic* does not contain `MessageDescription.Data.SimpleItemDescription` item with Name = "Failed", FAIL the test and skip other steps.
- 6.5. If *powerSupplyFailureTopic.MessageDescription.Data.SimpleItemDescription* item with Name = "Failed" does not have Type = "xs:boolean", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
- Filter.TopicExpression := "tns1:Device/HardwareFailure/PowerSupplyFailure"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
- SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
- 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
- Timeout := PT60S



- MessageLimit := 1

9.2. The DUT responds with **PullMessagesResponse** message with parameters

- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Device/HardwareFailure/PowerSupplyFailure, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Source.SimpleItem.Token, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Source.SimpleItem.Token has value type different from tt:ReferenceToken type, FAIL the test and go to the step 10.

9.3.4. If *m* does not contain Message.Message.Data.SimpleItem.Failed, FAIL the test and go to the step 10.

9.3.5. If *m*.Message.Message.Data.SimpleItem.Failed has value type different from xs:boolean type, FAIL the test and go to the step 10.

9.3.6. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

##### PASS –

The DUT passed all assertions.

##### FAIL –

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

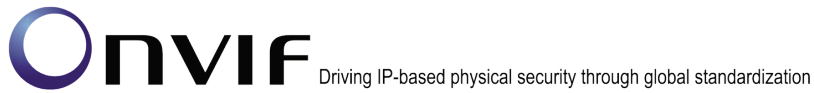
The DUT did not send **CreatePullPointSubscriptionResponse** message.

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.





#### **4.3.11 Storage Failure event**

**Test Label:** Storage Failure event

**Test Case ID:** QUICK\_INSTALL-3-1-11

**ONVIF Core Specification Coverage:** Monitoring Event Storage Failure

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Device/HardwareFailure/StorageFailure event

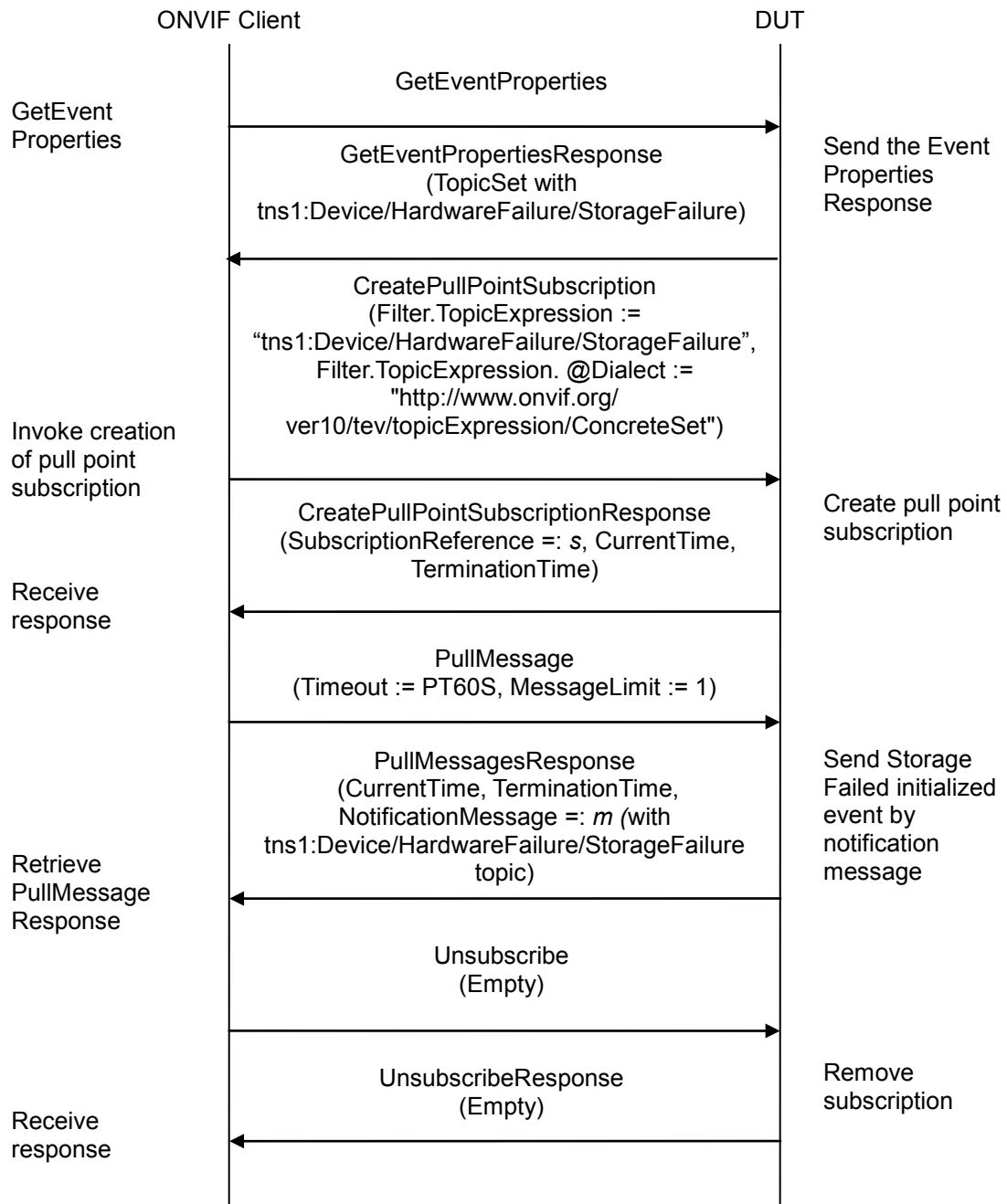
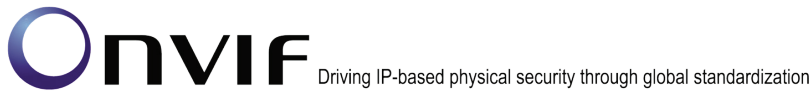
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Device/HardwareFailure/StorageFailure event generation and to verify tns1:Device/HardwareFailure/StorageFailure event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Device/HardwareFailure/StorageFailure event is supported by the DUT as indicated by the GetEventPropertiesResponse.

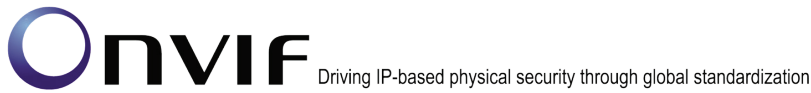
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**

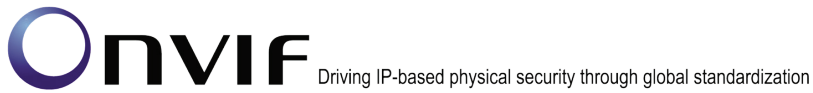


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters



- TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: topicSet
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain tns1:Device/HardwareFailure/StorageFailure topic, FAIL the test and skip other steps.
6. ONVIF Client verifies tns1:Device/HardwareFailure/StorageFailure topic (*storageFailureTopic*) from *topicSet*:
- 6.1. If *storageFailureTopic*.MessageDescription.IsProperty is skipped or equals to false, FAIL the test and skip other steps.
  - 6.2. If *storageFailureTopic* does not contain MessageDescription.Source.SimpleItemDescription item with Name = "Token", FAIL the test and skip other steps.
  - 6.3. If *storageFailureTopic*.MessageDescription.Source.SimpleItemDescription item with Name = "Token" does not have Type = "tt:ReferenceToken", FAIL the test and skip other steps.
  - 6.4. If *storageFailureTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Failed", FAIL the test and skip other steps.
  - 6.5. If *storageFailureTopic*.MessageDescription.Data.SimpleItemDescription item with Name = "Failed" does not have Type = "xs:boolean", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
- Filter.TopicExpression := "tns1:Device/HardwareFailure/StorageFailure"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
- SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
- 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
- Timeout := PT60S
  - MessageLimit := 1
- 9.2. The DUT responds with **PullMessagesResponse** message with parameters



- CurrentTime
- TerminationTime
- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Device/HardwareFailure/StorageFailure, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Source.SimpleItem.Token, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Source.SimpleItem.Token has value type different from tt:ReferenceToken type, FAIL the test and go to the step 10.

9.3.4. If *m* does not contain Message.Message.Data.SimpleItem.Failed, FAIL the test and go to the step 10.

9.3.5. If *m*.Message.Message.Data.SimpleItem.Failed has value type different from xs:boolean type, FAIL the test and go to the step 10.

9.3.6. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

##### PASS –

The DUT passed all assertions.

##### FAIL –

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

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

The DUT did not send **PullMessagesResponse** message(s).

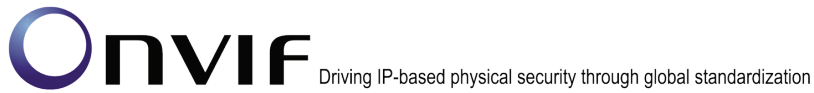
The DUT did not send **UnsubscribeResponse** message.

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

#### 4.3.12 Critical Temperature event

**Test Label:** Critical Temperature event

**Test Case ID:** QUICK\_INSTALL-3-1-12



**ONVIF Core Specification Coverage:** Monitoring Event Critical Temperature

**Command Under Test:** GetServices, GetEventProperties, CreatePullPointSubscription, PullMessages, Device/HardwareFailure/TemperatureCritical event

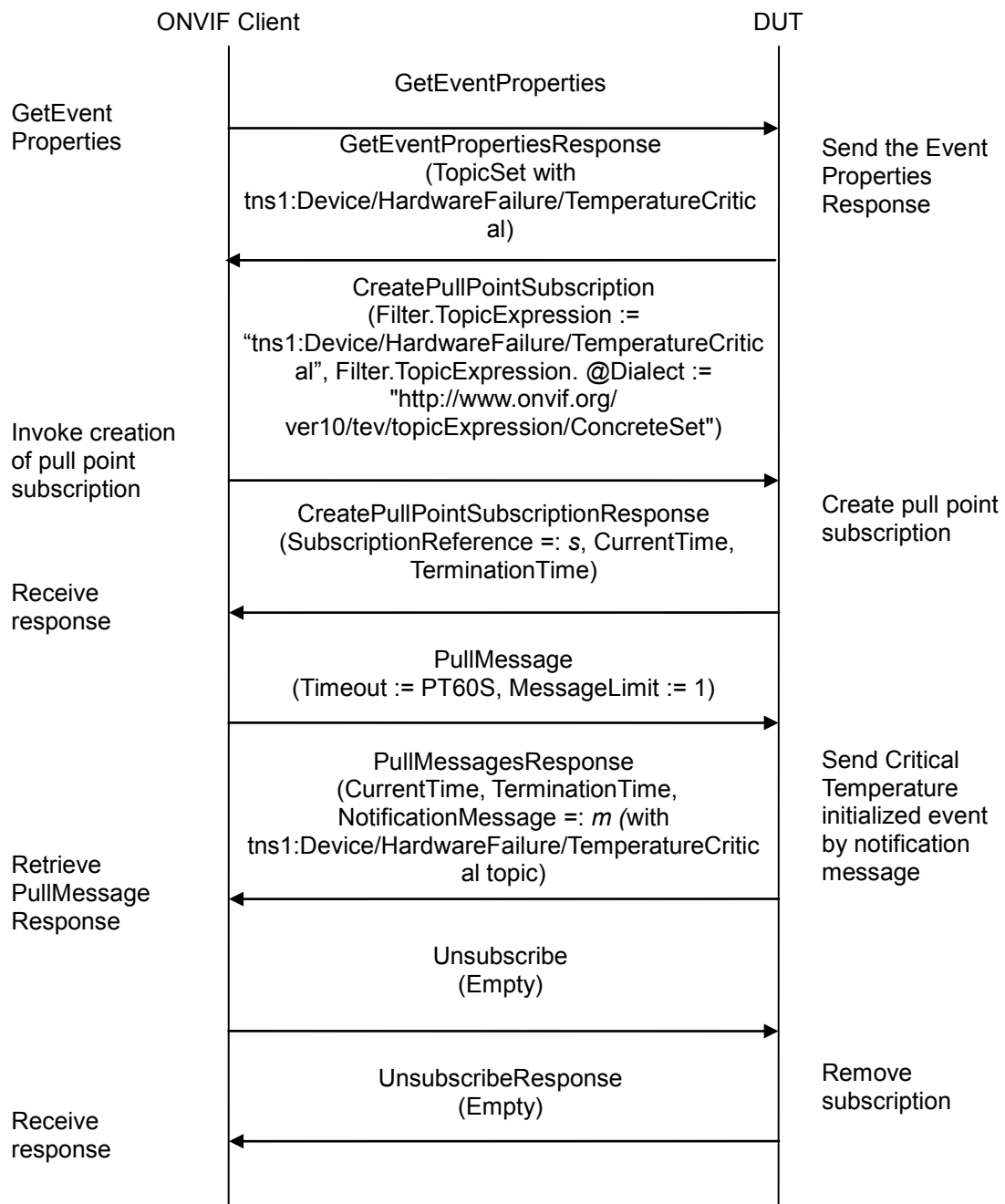
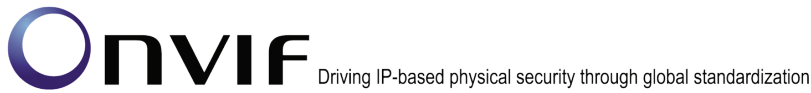
**WSDL Reference:** devicemgmt.wsdl and event.wsdl

**Test Purpose:** To verify tns1:Device/HardwareFailure/TemperatureCritical event generation and to verify tns1:Device/HardwareFailure/TemperatureCritical event format.

**Pre-requisite:** Event Service was received from the DUT. tns1:Device/HardwareFailure/TemperatureCritical event is supported by the DUT as indicated by the GetEventPropertiesResponse.

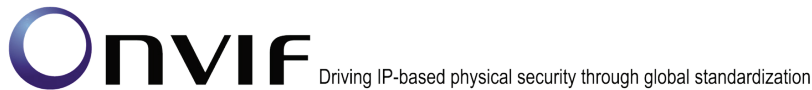
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**

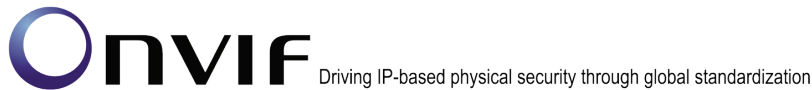


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetEventProperties**.
4. The DUT responds with a **GetEventPropertiesResponse** message with parameters



- TopicNamespaceLocation list
  - FixedTopicSet
  - TopicSet =: topicSet
  - TopicExpressionDialect list
  - MessageContentFilterDialect list
  - MessageContentSchemaLocation list
5. If *topicSet* does not contain `tns1:Device/HardwareFailure/TemperatureCritical` topic, FAIL the test and skip other steps.
6. ONVIF Client verifies `tns1:Device/HardwareFailure/TemperatureCritical` topic (*criticalTemperatureTopic*) from *topicSet*:
- 6.1. If *criticalTemperatureTopic*.MessageDescription.IsProperty is skipped or equals to false, FAIL the test and skip other steps.
- 6.2. If *criticalTemperatureTopic* does not contain MessageDescription.Data.SimpleItemDescription item with Name = "Critical", FAIL the test and skip other steps.
- 6.3. If *criticalTemperatureTopic*.MessageDescription.Data.SimpleItemDescription item with Name = "Critical" does not have Type = "xs:boolean", FAIL the test and skip other steps.
7. ONVIF Client invokes **CreatePullPointSubscription** with parameters
- Filter.TopicExpression := "tns1:Device/HardwareFailure/TemperatureCritical"
  - Filter.TopicExpression.@Dialect := "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
8. The DUT responds with a **CreatePullPointSubscriptionResponse** message with parameters
- SubscriptionReference =: s
  - CurrentTime
  - TerminationTime
9. Until *timeout1* timeout expires, repeat the following steps:
- 9.1. ONVIF Client invokes **PullMessages** to the subscription endpoint *s* with parameters
- Timeout := PT60S
  - MessageLimit := 1
- 9.2. The DUT responds with **PullMessagesResponse** message with parameters
- CurrentTime
  - TerminationTime



- NotificationMessage =: *m*

9.3. If *m* is not null and *m*.Message.Message.PropertyOperation="Initialized" ONVIF Client verifies *m*:

9.3.1. If *m*.Topic does not equal to tns1:Device/HardwareFailure/TemperatureCritical, FAIL the test and go to the step 10.

9.3.2. If *m* does not contain Message.Message.Data.SimpleItem.Critical, FAIL the test and go to the step 10.

9.3.3. If *m*.Message.Message.Data.SimpleItem.Critical has value type different from xs:boolean type, FAIL the test and go to the step 10.

9.3.4. Go to the step 10.

9.4. If *timeout1* timeout expires for step 9 without Notification with PropertyOperation="Initialized", FAIL the test and go to the step 10.

10. ONVIF Client sends an **Unsubscribe** to the subscription endpoint *s*.

11. The DUT responds with **UnsubscribeResponse** message.

#### Test Result:

##### PASS –

The DUT passed all assertions.

##### FAIL –

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

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

The DUT did not send **PullMessagesResponse** message(s).

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

**Note:** *timeout1* will be taken from the Operation Delay field of ONVIF Device Test Tool.

## 4.4 System

### 4.4.1 Firmware Upload

**Test Label:** System -FirmwareUpgrade

**Test Case ID:** QUICK\_INSTALL-4-1-1

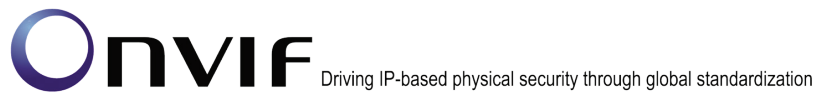
**ONVIF Core Specification Coverage:** Firmware Upload via HTTP

**Command Under Test:** StartFirmwareUpgrade

**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify that the firmware upgrade is correctly executed.

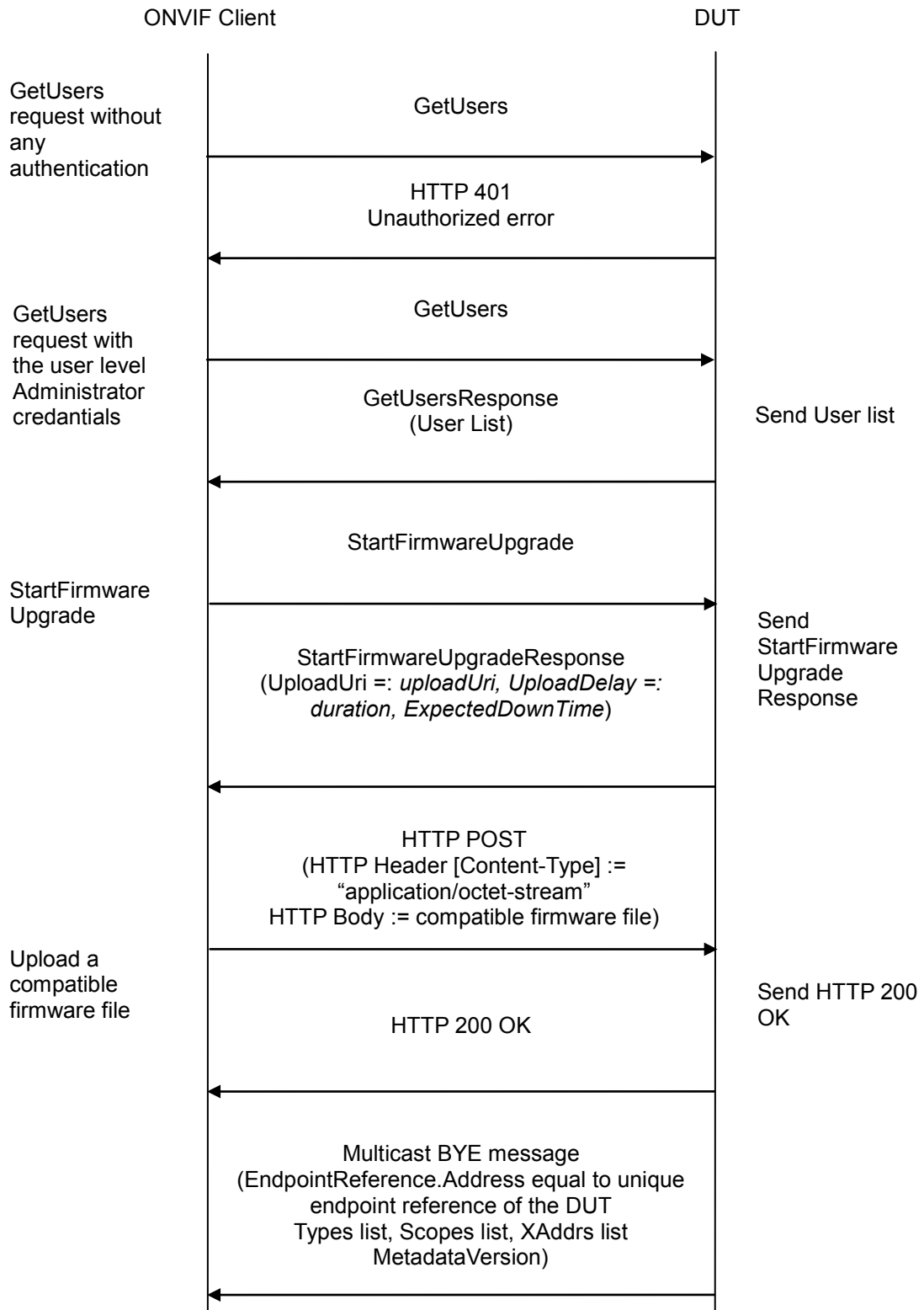
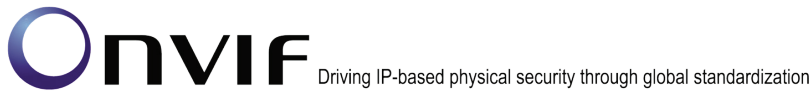


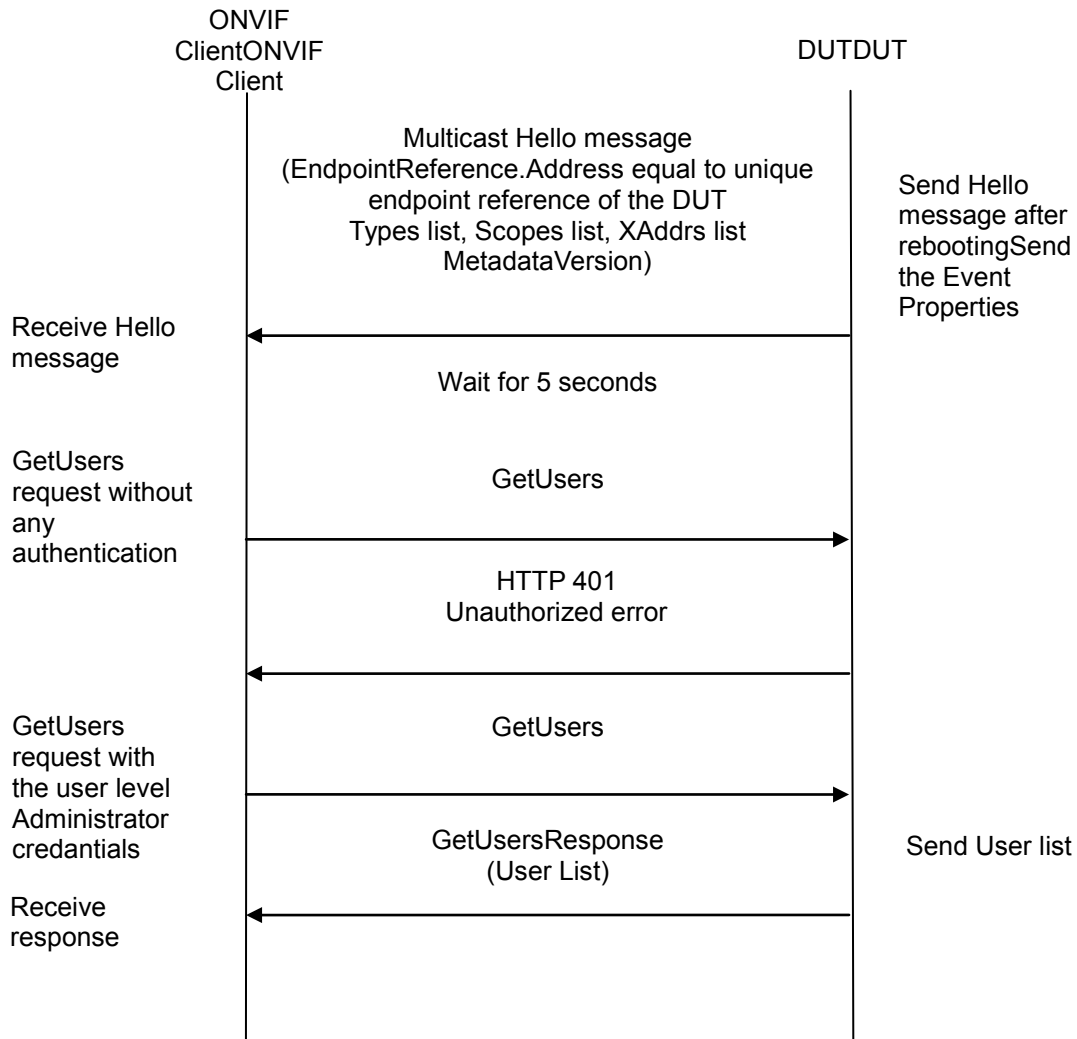
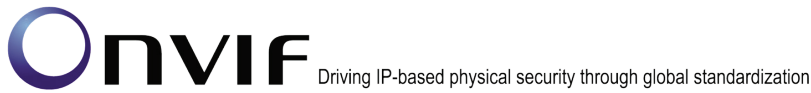


**Pre-requisite:** GetServices command is supported by the DUT. HttpFirmwareUpgrade feature is supported by the DUT as indicated by the System.HttpFirmwareUpgrade. Location of Compatible firmware file is provided.

**Test Configuration:** ONVIF Client and DUT

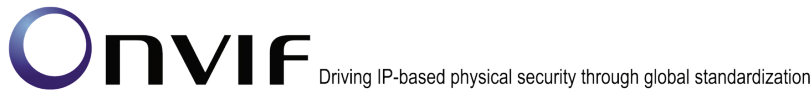
**Test Sequence:**



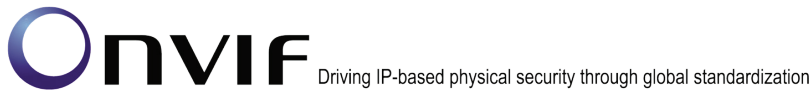


**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetUsers** without any authentication.
4. The DUT responds with **HTTP 401 Unauthorized** error.
5. ONVIF Client invokes **GetUsers** with user with the user level Administrator credentials.
6. The DUT responds with a **GetUsersResponse** message with parameters
  - UserList
7. ONVIF Client invokes **StartFirmwareUpgrade**.



8. The DUT responds with a **StartFirmwareUpgradeResponse** message with parameters
  - UploadUri =: *uploadUri*
  - UploadDelay =: *uploadDuration*
  - ExpectedDownTime =: *downTimeDuration*
9. ONVIF Client waits for time *uploadDuration*.
10. ONVIF Client invokes **HTTP POST** to *uploadUri* with parameters
  - HTTP Header [Content-Type] := “application/octet-stream”
  - HTTP Body := compatible firmware file
11. The DUT responds with **HTTP 200 OK** message.
12. If BYE message is supported by the DUT as indicated by the System.DiscoveryBye capabilities, the DUT will send Multicast **BYE** message before rebooting with parameters
  - EndpointReference.Address equal to unique endpoint reference of the DUT
  - Types list
  - Scopes list
  - XAddrs list
  - MetadataVersion
13. Until *timeout1* timeout expires, repeat the following steps:
  - 13.1. The DUT will send Multicast **Hello** message after it is successfully rebooted with parameters:
    - EndpointReference.Address equal to unique endpoint reference of the DUT
    - Types list
    - Scopes list
    - XAddrs list := *xaddrsList*
    - MetadataVersion
  - 13.2. If *xaddrsList* contains URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, go to step 15.
14. If *timeout1* timeout expires for step 5 without Hello with URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, FAIL the test and skip other steps.
15. ONVIF client waits for 5 seconds after Hello was received.
16. ONVIF Client invokes **GetUsers** without any authentication.
17. The DUT responds with **HTTP 401 Unauthorized** error.



18. ONVIF Client invokes **GetUsers** with user with the user level Administrator credentials.

19. The DUT responds with a **GetUsersResponse** message with parameters

- UserList

**Test Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

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

The DUT allowed Anonymous access to the **GetUsers** command.

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

The DUT did not response with **HTTP 200 OK** to HTTP POST request.

The DUT did not send **Bye** message with EndpointReference.Address equal to unique endpoint reference of the DUT in case Bye message was supported by the DUT.

**Note:** Reboot Timeout will be used for waiting for the Bye message from the DUT. Reboot Timeout will be taken from the Reboot Timeout field of ONVIF Device Test Tool.

**Note:** *timeout1* will be selected as maximum of (*downTimeDuration*, Reboot Timeout), where Reboot Timeout will be taken from the Reboot Timeout field of ONVIF Device Test Tool.

**Note:** IPv4 address from Hello shall be used for further test cases.

#### 4.4.2 Invalid Firmware Upload

**Test Label:** System -FirmwareUpgradeInvalid

**Test Case ID:** QUICK\_INSTALL-4-1-2

**ONVIF Core Specification Coverage:** Firmware Upload via HTTP

**Command Under Test:** StartFirmwareUpgrade

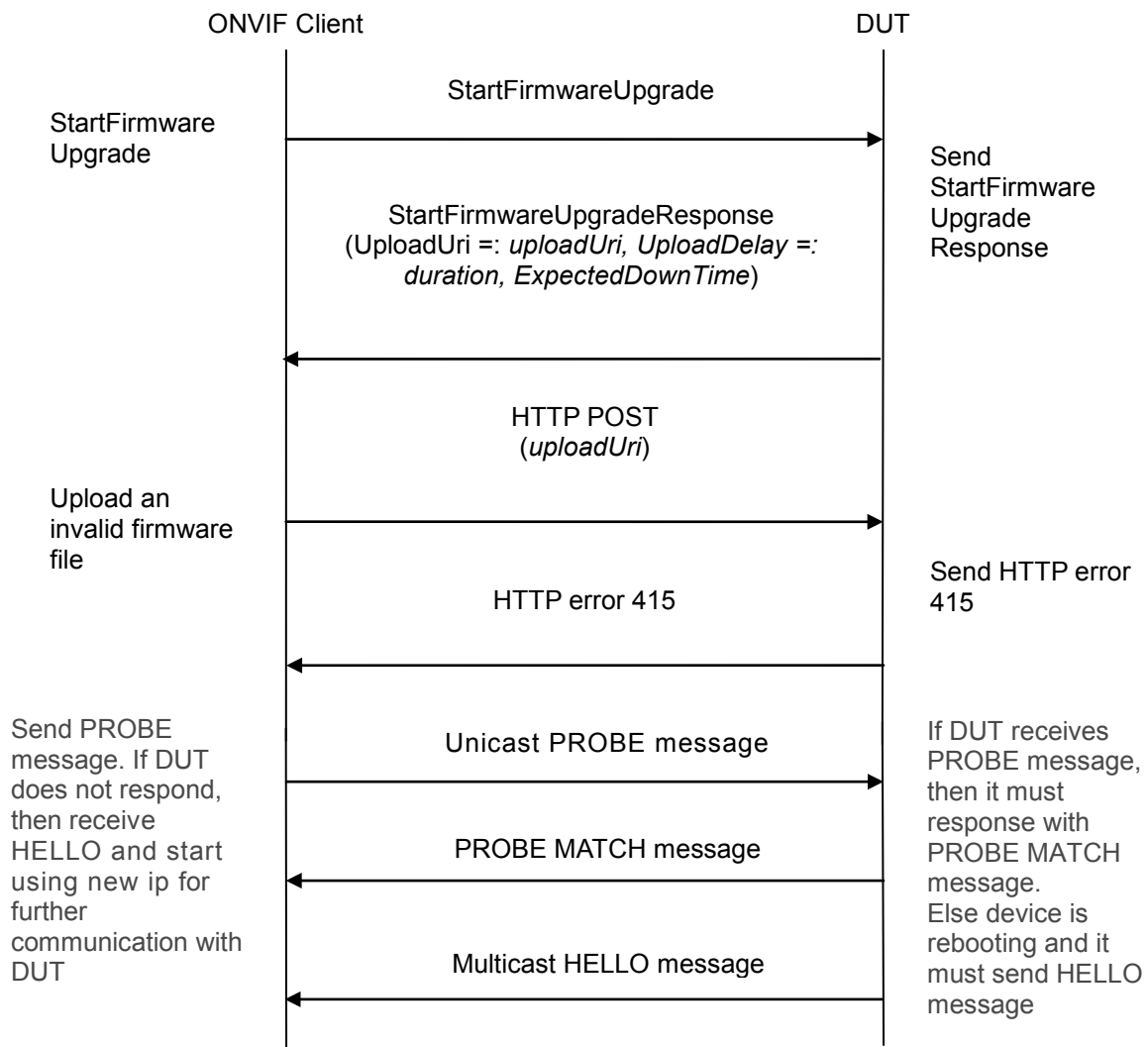
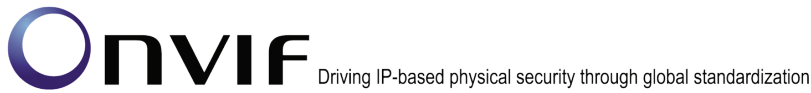
**WSDL Reference:** devicemgmt.wsdl

**Test Purpose:** To verify that the firmware upgrade fails with the expected error code.

**Pre-requisite:** GetServices command is supported by the DUT. HttpFirmwareUpgradefeature is supported by the DUT as indicated by the System.HttpFirmwareUpgrade.

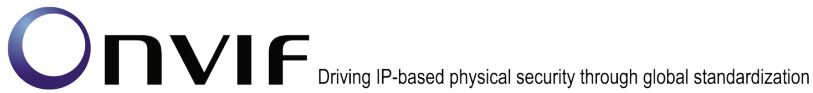
**Test Configuration:** ONVIF Client and DUT

**Test Sequence:**



**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **StartFirmwareUpgrade**.
4. The DUT responds with a **StartFirmwareUpgradeResponse** message with parameters
  - UploadUri =: *uploadUri*
  - UploadDelay =: *duration*
  - ExpectedDownTime
5. ONVIF Client waits for time *duration*.



6. ONVIF Client invokes **HTTP POST** to *uploadUri* with parameters
  - HTTP Header [Content-Type] := “application/octet-stream”
  - HTTP Body := invalid firmware file
7. The DUT responds with **HTTP 415** message.
8. ONVIF client waits Reboot timeout.
9. ONVIF Client sends PROBE message and if DUT does not respond with PROBE MATCH message then go to the step 10. If DUT responds, then finish the test.
10. ONVIF Client waits for Hello message sent from newly configured address by the DUT. Then ONVIF Client starts using this newly configured address for further communications with DUT.

**Test Result:**

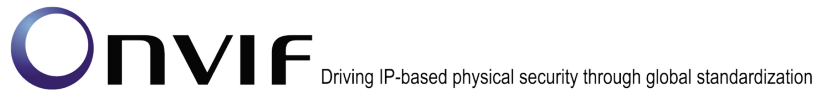
**PASS –**

The DUT passed all assertions.

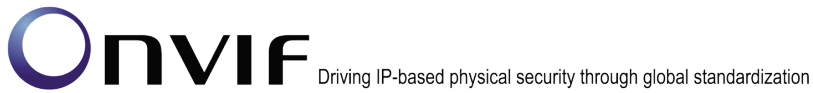
**FAIL –**

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

The DUT did not response with **HTTP 415** to HTTP POST request.







## Annex A

This section describes the meaning of the following definitions. These definitions are used in the test case description.

### **A.1 Create user with defined user level**

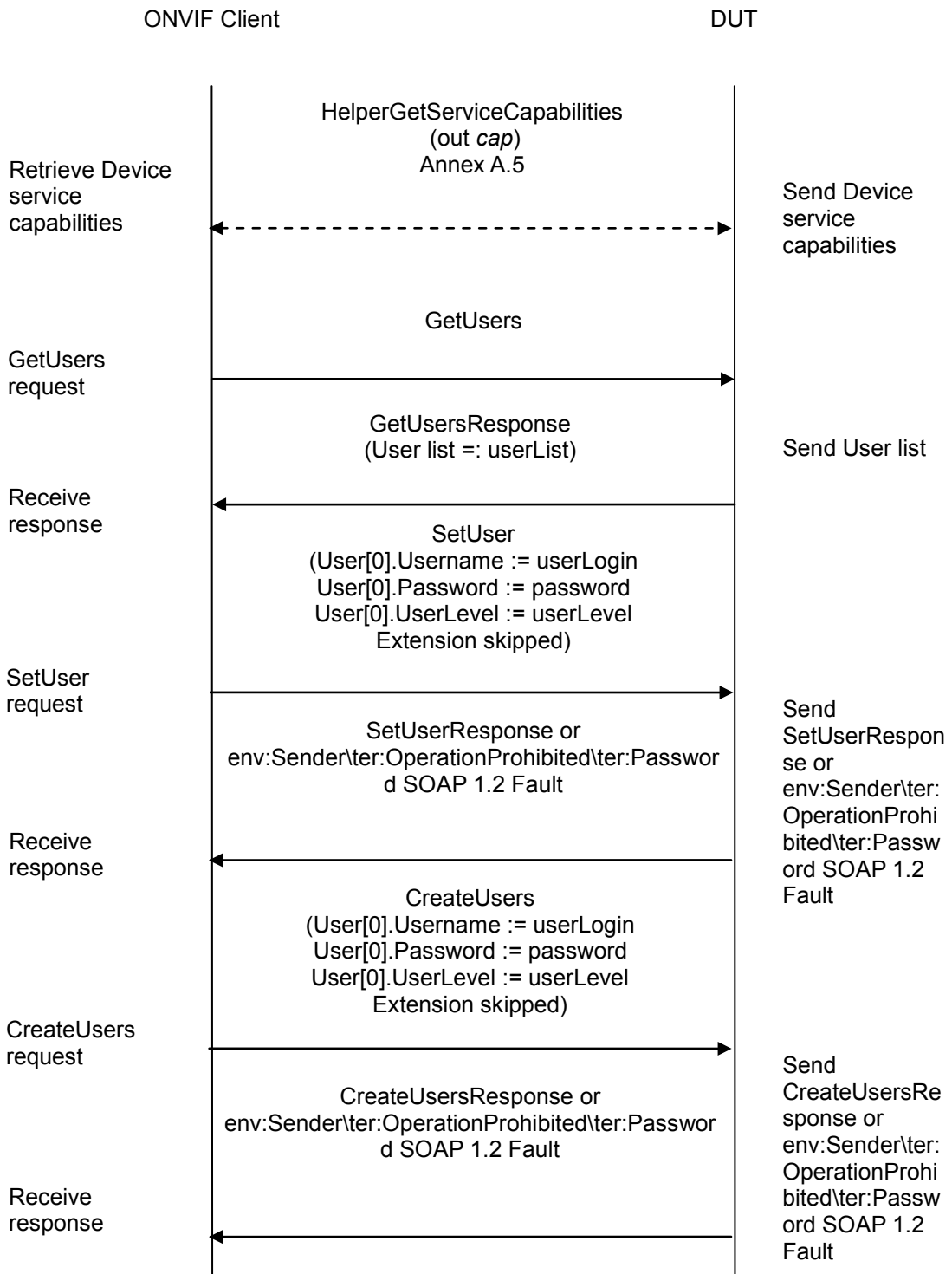
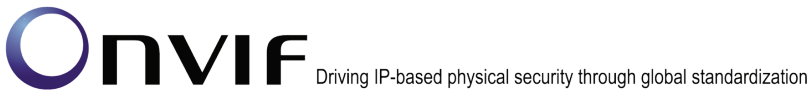
**Name:** HelperCreateUserLevel

**Procedure Purpose:** Helper procedure to create user with predefined user level or change existing with new one.

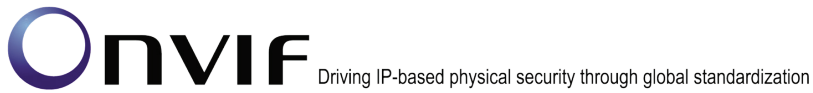
**Pre-requisite:** GetServices command is supported by the DUT. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security.MaxUsernameLength, Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security.MaxPasswordLength.

**Input:** The user level (*userLevel*) of user to be created (*userLevel* shall have User or Operator value).

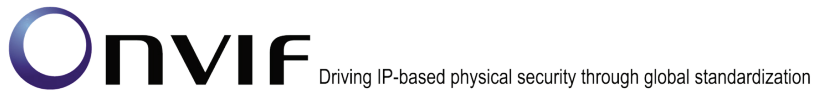
**Returns:** The user login (*userLogin*) with predefined user level and corresponding user password (*password*).



**Procedure:**



1. ONVIF Client gets the service capabilities (out *cap*) by following the procedure mentioned in Annex A.5.
2. If *cap* does not contain *Security.MaxPasswordLength* or *Security.MaxUserNameLength*, FAIL the test and skip other steps.
3. ONVIF Client invokes **GetUsers**.
4. The DUT responds with a **GetUsersResponse** message with parameters
  - User list := *userList*
5. If there is user with user level *userLevel* in *userList*:
  - 5.1. Set the following:
    - *passwordLength* := *cap.Security.MaxPasswordLength*
    - *userLogin* := Username of user with user level equal to *userLevel* from *userList*
    - *password* := random string, contains *passwordLength* ASCII characters
  - 5.2. ONVIF Client invokes **SetUser** with parameters
    - User[0].Username := *userLogin*
    - User[0].Password := *password*
    - User[0].UserLevel := *userLevel*
    - Extension skipped
  - 5.3. If the DUT responds with **SetUserResponse** message, skip other steps.
  - 5.4. If the DUT returns env:Sender\ter:OperationProhibited\ter>Password SOAP 1.2 fault:
    - 5.4.1. Set the following:
      - *password* := random string, contains *passwordLength* ASCII characters
    - 5.4.2. Go to the step 6.2.
  - 5.5. If DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.
6. If there are no users with user level *userLevel* in *userList*:
  - 6.1. Set the following:
    - *userLoginLength* := *cap.Security.MaxUserNameLength*
    - *passwordLength* := *cap.Security.MaxPasswordLength*
    - *userLogin* := random string, contains *userLoginLength* low case alphabet characters, differs from usernames listed in *userList*
    - *password* := random string, contains *passwordLength* ASCII characters
  - 6.2. ONVIF Client invokes **CreateUsers** with parameters



- User[0].Username := *userLogin*
- User[0].Password := *password*
- User[0].UserLevel := *userLevel*
- Extension skipped

6.3. If the DUT responds with **CreateUsersResponse** message, skip other steps.

6.4. If the DUT returns env:Sender\ter:OperationProhibited\ter>Password SOAP 1.2 fault:

6.4.1. Set the following:

- *password* := random string, contains *passwordLength* ASCII characters

6.4.2. Go to the step 7.2.

6.4.3. If the DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.

**Procedure Result:**

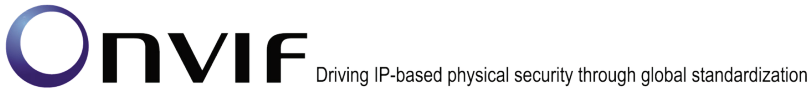
**PASS –**

The DUT passed all assertions.

**FAIL –**

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

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



**A.2 Restoring System Date and Time**

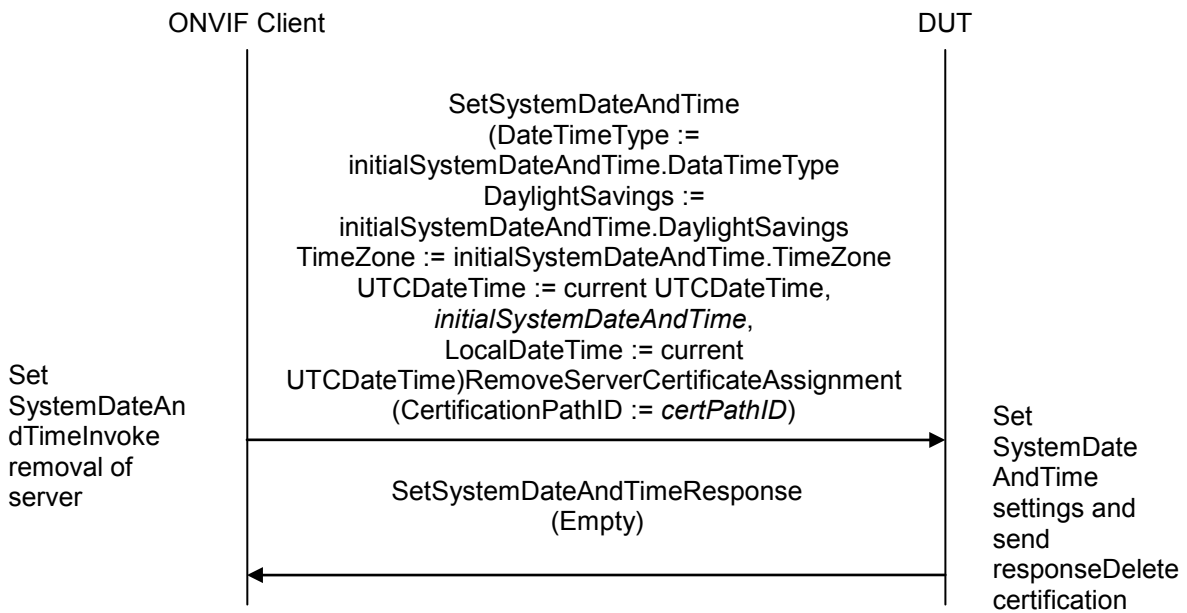
**Name:** HelperRestoreSystemDateAndTime

**Procedure Purpose:** Helper procedure to restoring System Date and Time.

**Pre-requisite:** None.

**Input:** The SystemDateAndTime (*initialSystemDateAndTime*) to restore.

**Returns:** None.

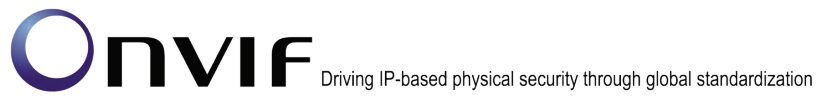


**Procedure:**

1. ONVIF Client invokes **SetSystemDateAndTime** with parameters
  - DateTimeType := *initialSystemDateAndTime.DateTimeType*
  - DaylightSavings := *initialSystemDateAndTime.DaylightSavings*
  - TimeZone := *initialSystemDateAndTime.TimeZone*
  - If *initialSystemDateAndTime.UTCDateTime* is not null UTCDateTime := current UTCDateTime
  - If *initialSystemDateAndTime.LocalDateTime* is not null LocalDateTime := current UTCDateTime
2. The DUT responds with a **SetSystemDateAndTimeResponse** message.

**Procedure Result:**

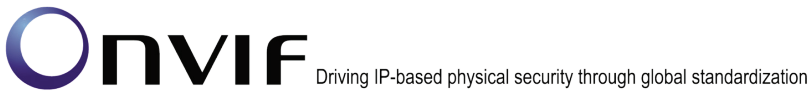
**PASS –**



The DUT passed all assertions.

**FAIL –**

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



### A.3 Set NTP settings

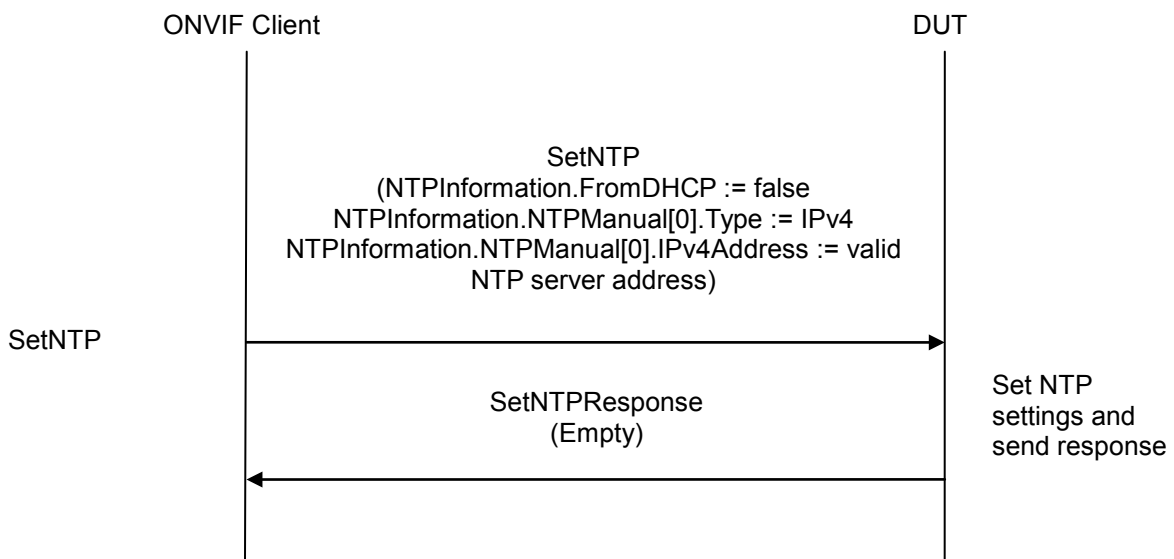
**Name:** HelperSetNTP

**Procedure Purpose:** Helper procedure to configure DUT with proper NTP server.

**Pre-requisite:** NTP is supported by the DUT as indicated by the Network.NTP capability. A valid NTP server address should be configured in the DUT.

**Input:** None.

**Returns:** None.



**Procedure:**

1. ONVIF Client invokes **SetNTP** with parameters
  - NTPInformation.FromDHCP := false
  - NTPInformation.NTPManual[0].Type := IPv4
  - NTPInformation.NTPManual[0].IPv4Address := valid NTP server address
2. The DUT responds with a **SetNTPResponse** message.

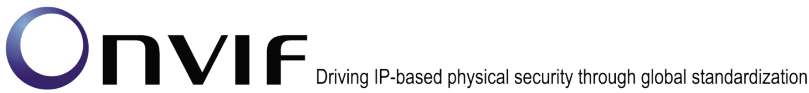
**Procedure Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

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



### A.4 Restoring NTP settings

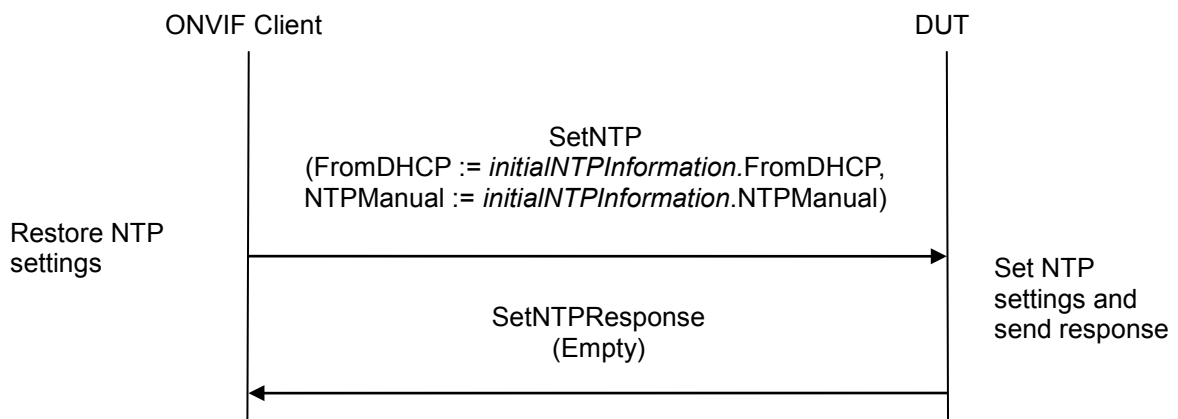
**Name:** HelperRestoreNTP

**Procedure Purpose:** Helper procedure to restoring NTP settings.

**Pre-requisite:** NTP is supported by the DUT as indicated by the Network.NTP capability.

**Input:** The initial NTP settings (*initialNTPInformation*) to restore.

**Returns:** None.



**Procedure:**

1. ONVIF Client invokes **SetNTP** with parameters
  - FromDHCP := *initialNTPInformation.FromDHCP*
  - NTPManual := *initialNTPInformation.NTPManual*
2. The DUT responds with a **SetNTPResponse** message.

**Procedure Result:**

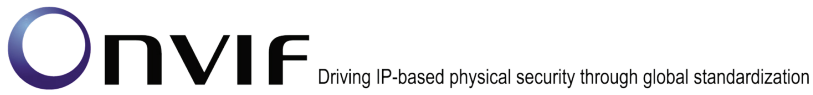
**PASS –**

The DUT passed all assertions.

**FAIL –**

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





### A.5 Get service capabilities

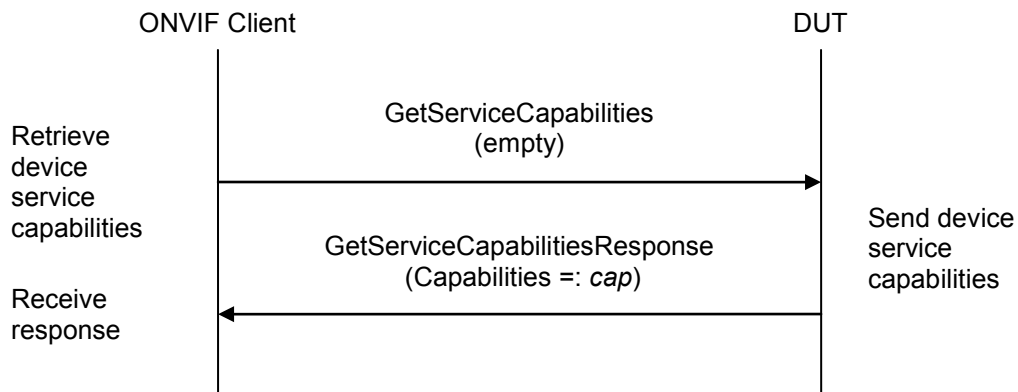
**Name:** HelperGetServiceCapabilities

**Procedure Purpose:** Helper procedure to get device service capabilities.

**Pre-requisite:** None.

**Input:** None

**Returns:** The service capabilities (*cap*).



**Procedure:**

1. ONVIF Client invokes **GetServiceCapabilities**.
2. The DUT responds with a **GetServiceCapabilitiesResponse** message with parameters
  - Capabilities =: *cap*

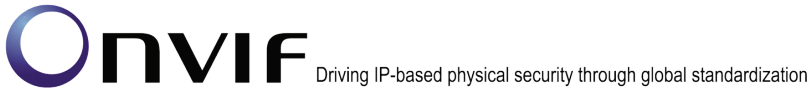
**Procedure Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

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



### A.6 Time synchronization

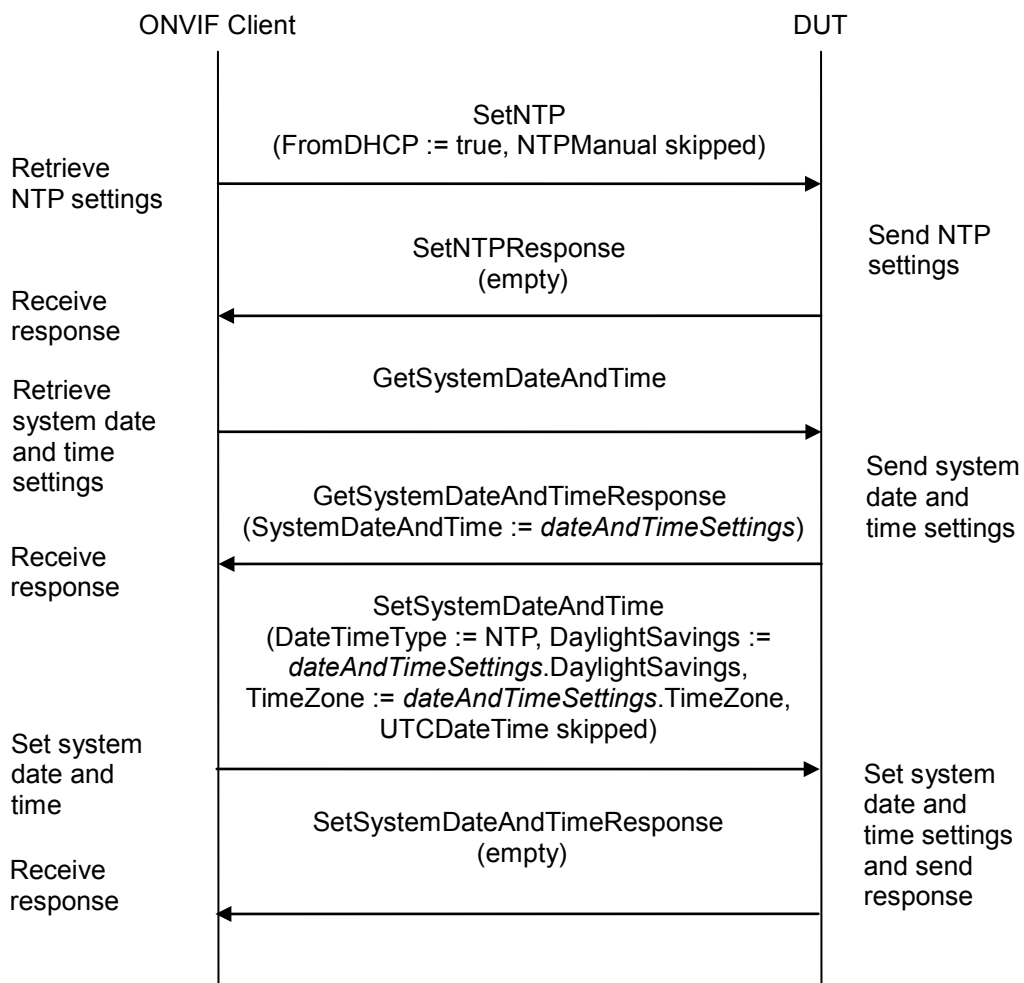
**Name:** HelperTimeSynchronization

**Procedure Purpose:** Helper procedure for time synchronization between ONVIF Client and DUT after hard system factory default.

**Pre-requisite:** None.

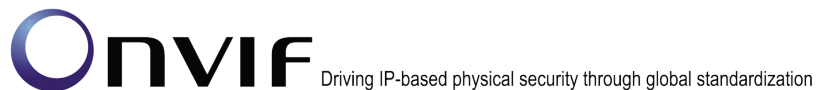
**Input:** None

**Returns:** None



**Procedure:**

1. ONVIF Client invokes **SetNTP** without any authentication with parameters
  - FromDHCP := true
  - NTPManual skipped



2. The DUT responds with a **SetNTPResponse** message.
3. ONVIF client invokes **GetSystemDateAndTime** without any authentication.
4. The DUT responds with **GetSystemDateAndTimeResponse** message with parameters
  - `SystemDateAndTime := dateAndTimeSettings`
5. ONVIF Client invokes **SetSystemDateAndTime** without any authentication with parameters
  - `DateTimeType := NTP`
  - `DaylightSavings := dateAndTimeSettings.DaylightSavings`
  - `TimeZone := dateAndTimeSettings.TimeZone`
  - UTCDateTime skipped
6. The DUT responds with a **SetSystemDateAndTime** message.

**Procedure Result:**

**PASS –**

The DUT passed all assertions.

**FAIL –**

The DUT did not allow Anonymous access to the **SetNTP** command.

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

The DUT did not allow Anonymous access to the **GetSystemDateAndTime** command.

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

The DUT did not allow Anonymous access to the **SetSystemDateAndTime** command.

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