ONVIF®

Real Time Streaming Device Test Specification

Version 24.12

December 2024

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

Vers.	Date	Description
13.06	June, 2013	First issue. The document was created by splitting Media Test Specification. The following test cases were added or updated with ID change: MEDIA CONTROL – RTSP/TCP MEDIA STREAMING – RTSP KEEPALIVE (SET_PARAMETER) MEDIA STREAMING - RTSP KEEPALIVE (OPTIONS) MEDIA STREAMING – JPEG (RTP-Unicast / UDP) MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – JPEG (RTP/RTSP/TCP) MEDIA STREAMING – MPEG4 (RTP-Unicast/ UDP) MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP) SET SYNCHRONIZATION POINT – MPEG4 MEDIA STREAMING – H.264 (RTP-Unicast/ UDP) MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – H.264 (RTP/RTSP/TCP) SET SYNCHRONIZATION POINT – H.264 MEDIA STREAMING – RTP-Unicast/RTSP/HTTP/TCP (LINE BREAKS IN BASE64 ENCODING) VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION (ALL RESOLUTIONS) VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION (ALL RESOLUTIONS) VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION (ALL RESOLUTIONS) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS)

		MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) (ALL VIDEO SOURCE CONFIGURATIONS) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) MEDIA STREAMING – JPEG (VALIDATING RTP HEADER EXTENSION) MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv6) MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv6) MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv6) VIDEO ENCODER CONFIGURATION – MULTICAST PORT (IPv4) VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4) VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv4) VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv6) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) MEDIA STREAMING – G.711 (RTP-Unicast/UDP) MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – G.711 (RTP/RTSP/TCP) MEDIA STREAMING – G.726 (RTP-Unicast/UDP) MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – G.726 (RTP/RTSP/TCP)

		MEDIA STREAMING – AAC (RTP-Unicast/UDP) MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – AAC (RTP/RTSP/TCP) AUDIO STREAMING – G.711 (RTP-Unicast/UDP) AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP) AUDIO STREAMING – G.711 (RTP/RTSP/TCP) AUDIO STREAMING – G.726 (RTP-Unicast/UDP) AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP) AUDIO STREAMING – G.726 (RTP/RTSP/TCP) AUDIO STREAMING – AAC (RTP-Unicast/UDP) AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) AUDIO STREAMING – AAC (RTP/RTSP/TCP) AUDIO ENCODER CONFIGURATION – MULTICAST PORT (IPv4) AUDIO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4) MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv6) MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv6) MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/ UDP) MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP) MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/ UDP) MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP) MEDIA STREAMING – JPEG/AAC (RTP-Unicast/ UDP) MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP) MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP) MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv6)

		MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv6) MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv4) MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv6) VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT PORTS VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT ADDRESS NOTIFICATION STREAMING START AND STOP MULTICAST STREAMING – JPEG (IPv4) START AND STOP MULTICAST STREAMING – G.711 (IPv4) START AND STOP MULTICAST STREAMING – JPEG/G.711 (IPv4) START AND STOP MULTICAST STREAMING – G.711 (IPv4, ONLY AUDIO PROFILE) START AND STOP MULTICAST STREAMING – G.726 (IPv4, ONLY AUDIO PROFILE) START AND STOP MULTICAST STREAMING – AAC (IPv4, ONLY AUDIO PROFILE) The following Annexes were added or updated: Annex A.9 Create Empty Profile Annex A.10 Find or Create Media Profile Containing Specified Audio Encoder Configuration Annex A.11 Configure Multicast Settings for Configurations that are Included in Profile Annex A.12 Find or Create Media Profile Containing Specified Video Encoder Configuration Annex A.13 Find or Create Media Profile Containing Specified Video Encoder Configuration with Possibility to Set Specified Encoding Annex A.14 Find or Create Specified Number of Profiles that Contains Specified Video Source Configuration and Video Encoder Configuration Annex A.20 Media Profile Select or Creation for Audio and Video Streaming Annex A.21 Media Profile Creation for Video Streaming Annex A.22 Name and Token Parameters Maximum Length
13.12	December, 2013	The following test cases were added: MEDIA STREAMING – JPEG (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG (RTP/RTSP/TCP, IPv6)

		MEDIA STREAMING – MPEG4 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – H.264 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – H.264 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – G.711 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – G.711 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – G.726 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – G.726 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – AAC (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – AAC (RTP/RTSP/TCP, IPv6) AUDIO STREAMING – G.711 (RTP-Unicast/UDP, IPv6) AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) AUDIO STREAMING – G.711 (RTP/RTSP/TCP, IPv6) AUDIO STREAMING – G.726 (RTP-Unicast/UDP, IPv6) AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) AUDIO STREAMING – G.726 (RTP/RTSP/TCP, IPv6) AUDIO STREAMING – AAC (RTP-Unicast/UDP, IPv6) AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) AUDIO STREAMING – AAC (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP, IPv6)

		MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP, IPv6) The following Annexes were added: A.23 Turn on IPv6 network interface A.24 Restore Network Settings
13.12	December, 2013	The pre-requisites were updated for all test cases The following test cases were added or changed: VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION MEDIA STREAMING – JPEG (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – MPEG4 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – H.264 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – H.264 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – G.711 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – G.711 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – G.726 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – G.726 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – AAC (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – AAC (RTP/RTSP/TCP, IPv6) AUDIO STREAMING – G.711 (RTP-Unicast/UDP, IPv6) AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) AUDIO STREAMING – G.711 (RTP/RTSP/TCP, IPv6) AUDIO STREAMING – G.726 (RTP-Unicast/UDP, IPv6)

		AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) AUDIO STREAMING – G.726 (RTP/RTSP/TCP, IPv6) AUDIO STREAMING – AAC (RTP-Unicast/UDP, IPv6) AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) AUDIO STREAMING – AAC (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP, IPv6) MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP, IPv6) MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP, IPv6)
15.06	June, 2015	The following test cases were added: BACKCHANNEL – G.711 (RTP-Unicast/UDP) IPv4 BACKCHANNEL – G.711 (RTP-Unicast/RTSP/HTTP/TCP) IPv4 BACKCHANNEL – G.711 (RTP/RTSP/TCP) IPv4 BACKCHANNEL – G.726 (RTP- Unicast /UDP) IPv4 BACKCHANNEL – G.726 (RTP- Unicast/RTSP/HTTP/TCP) IPv4 BACKCHANNEL – G.726 (RTP/RTSP/TCP) IPv4 BACKCHANNEL – AAC (RTP-Unicast/UDP) IPv4 BACKCHANNEL – AAC (RTP-Unicast/RTSP/HTTP/TCP) IPv4 BACKCHANNEL – AAC (RTP/RTSP/TCP) IPv4 BACKCHANNEL – G.711 (RTP-Multicast/UDP) IPv4 BACKCHANNEL – G.726 (RTP-Multicast/UDP) IPv4 BACKCHANNEL – AAC (RTP-Multicast/UDP) IPv4 A.25 Media Profile Configuration for Backchannel Audio Streaming
16.01	January, 2016	The note to take into account the optional elements (number of profiles for JPEG, H264 or MPEG) has been removed from A.14
17.01	December, 2017	Minor changes: references to Annexes were updated MEDIA CONTROL – RTSP/TCP test case was updated

17.12	August 31, 2017	The following Annex was added according to #1285: RTSP Authentication Check
17.12	October 04, 2017	The following test cases were removed according to #1474: BACKCHANNEL – G.711 (RTP-Multicast/UDP, IPv4) BACKCHANNEL – G.726 (RTP-Multicast/UDP, IPv4) BACKCHANNEL – AAC (RTP-Multicast/UDP, IPv4)
18.06	June, 2018	Reformatting document using new template
18.12	Dec 21, 2018	Switching Hub description in 'Network Configuration for DUT' section was updated according to #1737
18.12	Dec 24, 2018	The following test cases were added according to #1697: BACKCHANNEL STREAMING BY POST – G.711 (RTP-Unicast/ RTSP/HTTP/TCP, IPv4) BACKCHANNEL STREAMING BY POST – G.726 (RTP-Unicast/ RTSP/HTTP/TCP, IPv4) BACKCHANNEL STREAMING BY POST – AAC (RTP-Unicast/ RTSP/HTTP/TCP, IPv4) The following test cases were changed to be consistency with implementation according to #1697: BACKCHANNEL – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) BACKCHANNEL – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) BACKCHANNEL – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4)
19.06	Feb 04, 2019	The following annexes were added according to #1773: The following annexes were added according to #1773: Annex A.27 Video Encoder Configuration – Resolution Selection (added) Annex A.28 Video Encoder Configuration – Frame Rate Selection (added) Note with link to Annex A.28 was added according to #1773 to the following test cases: RTSS-1-1-24 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION (ALL RESOLUTIONS) RTSS-1-1-25 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION (ALL RESOLUTIONS) RTSS-1-1-26 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION (ALL RESOLUTIONS) RTSS-1-1-46 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION RTSS-1-1-47 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION

			RTSS-1-1-48 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION RTSS-1-1-53 MEDIA STREAMING – JPEG (VALIDATING RTP HEADER EXTENSION) Note with link to Annex A.27 and Annex A.28 was added according to #1773 to the following test cases: RTSS-1-1-27 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP) RTSS-1-1-28 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-29 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) RTSS-1-1-30 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) RTSS-1-1-31 MEDIA CONTROL – RTSP/TCP RTSS-1-1-32 MEDIA STREAMING – RTSP KEEPALIVE (SET_PARAMETER) RTSS-1-1-33 MEDIA STREAMING - RTSP KEEPALIVE (OPTIONS) RTSS-1-1-34 MEDIA STREAMING – JPEG (RTP-Unicast / UDP) RTSS-1-1-35 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-36 MEDIA STREAMING – JPEG (RTP/RTSP/TCP) RTSS-1-1-37 MEDIA STREAMING – MPEG4 (RTP-Unicast/ UDP) RTSS-1-1-38 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-39 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP) RTSS-1-1-40 SET SYNCHRONIZATION POINT – MPEG4 RTSS-1-1-41 MEDIA STREAMING – H.264 (RTP-Unicast/ UDP) RTSS-1-1-42 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-43 MEDIA STREAMING – H.264 (RTP/RTSP/TCP) RTSS-1-1-44 SET SYNCHRONIZATION POINT – H.264 RTSS-1-1-45 MEDIA STREAMING – RTP-Unicast/RTSP/HTTP/TCP (LINE BREAKS IN BASE64 ENCODING) RTSS-1-1-49 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-1-1-50 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS)

		RTSS-1-1-51 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-1-1-52 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-1-1-54 MEDIA STREAMING – JPEG (RTP-Unicast/UDP, IPv6) RTSS-1-1-55 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-1-1-56 MEDIA STREAMING – JPEG (RTP/RTSP/TCP, IPv6) RTSS-1-1-57 MEDIA STREAMING – MPEG4 (RTP-Unicast/UDP, IPv6) RTSS-1-1-58 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-1-1-59 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP, IPv6) RTSS-1-1-60 MEDIA STREAMING – H.264 (RTP-Unicast/UDP, IPv6) RTSS-1-1-61 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-1-1-62 MEDIA STREAMING – H.264 (RTP/RTSP/TCP, IPv6) RTSS-1-2-12 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) RTSS-1-2-13 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4) RTSS-1-2-14 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4) RTSS-1-2-15 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4) RTSS-1-2-16 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv6) RTSS-1-2-17 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv6) RTSS-1-2-18 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv6) RTSS-1-2-19 VIDEO ENCODER CONFIGURATION – MULTICAST PORT (IPv4) RTSS-1-2-20 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4) RTSS-1-2-21 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv4) RTSS-1-2-22 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv6)

		RTSS-1-2-23 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-3-1-10 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/UDP) RTSS-3-1-11 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-3-1-12 MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP) RTSS-3-1-13 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/UDP) RTSS-3-1-14 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-3-1-15 MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP) RTSS-3-1-16 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP) RTSS-3-1-17 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP) RTSS-3-1-18 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP) RTSS-3-1-19 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/UDP, IPv6) RTSS-3-1-20 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-3-1-21 MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP, IPv6) RTSS-3-1-22 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/UDP, IPv6) RTSS-3-1-23 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-3-1-24 MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP, IPv6) RTSS-3-1-25 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP, IPv6) RTSS-3-1-26 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-3-1-27 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP, IPv6) RTSS-3-2-16 MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv4) RTSS-3-2-17 MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv6) RTSS-3-2-18 MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv4)

		RTSS-3-2-19 MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv6) RTSS-3-2-20 MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv4) RTSS-3-2-21 MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv6) RTSS-3-2-22 VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT PORTS RTSS-3-2-23 VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT ADDRESS RTSS-5-1-7 START AND STOP MULTICAST STREAMING – JPEG (IPv4) RTSS-5-1-9 START AND STOP MULTICAST STREAMING – JPEG/G.711 (IPv4)
19.06	Mar 25, 2019	The following annex was added according to #1803: Annex A.27 Removing Configurations from Media Profile The following annex was added according to #1803: MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) (step 8 added, step 15 added) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) (ALL VIDEO SOURCE CONFIGURATIONS) (step 8 added, step 15 added) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) (step 8 added, step 27 added) MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4) (step 6 added, step 21 added) MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4) (step 6 added, step 21 added) MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4) (step 6 added, step 21 added) MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv6) (step 6 added, step 21 added) MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv6) (step 6 added, step 21 added) MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv6) (step 6 added, step 21 added) VIDEO ENCODER CONFIGURATION – MULTICAST PORT (IPv4) (step 11 added, step 21 updated) VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4) (step 11 added, step 21 updated)

		VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv4) (step 6 added, step 21 added) VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv6) (step 6 added, step 21 added) MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) (step 8 added, step 26 added) MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv4) (step 8 added, step 23 added) MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv6) (step 8 added, step 23 added) MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv4) (step 8 added, step 23 added) MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv6) (step 8 added, step 23 added) MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv4) (step 8 added, step 23 added) MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv6) (step 8 added, step 23 added) VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT PORTS (step 12 added, step 22 added) VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT ADDRESS (step 12 added, step 22 added) START AND STOP MULTICAST STREAMING – JPEG (IPv4) (step 6 added, step 15 added) START AND STOP MULTICAST STREAMING – G.711 (IPv4) (step 6 added, step 15 added) START AND STOP MULTICAST STREAMING – JPEG/G.711 (IPv4) (step 8 added, step 17 added)
19.06	Apr 11, 2019	The following annex was added according to #1764: RTSS-2-1-25 MEDIA STREAMING – AAC (RTP-Unicast/UDP) (step 11 added) RTSS-2-1-26 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) (step 11 added) RTSS-2-1-27 MEDIA STREAMING – AAC (RTP/RTSP/TCP) (step 11 added) RTSS-2-1-34 AUDIO STREAMING – AAC (RTP-Unicast/UDP) (step 23 added) RTSS-2-1-35 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) (step 23 added) RTSS-2-1-36 AUDIO STREAMING – AAC (RTP/RTSP/TCP) (step 23 added) RTSS-2-1-43 MEDIA STREAMING – AAC (RTP-Unicast/UDP, IPv6) (step 12 added)

		RTSS-2-1-44 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) (step 12 added) RTSS-2-1-45 MEDIA STREAMING – AAC (RTP/RTSP/TCP, IPv6) (step 12 added) RTSS-2-1-52 AUDIO STREAMING – AAC (RTP-Unicast/UDP, IPv6) (step 24 added) RTSS-2-1-53 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) (step 24 added) RTSS-2-1-54 AUDIO STREAMING – AAC (RTP/RTSP/TCP, IPv6) (step 24 added) RTSS-2-2-15 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv4) (step 23 added) RTSS-2-2-16 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv6) (step 23 added) RTSS-3-1-16 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP) (step 13 and step 14 added) RTSS-3-1-17 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP) (step 15 and step 16 added) RTSS-3-1-18 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP) (step 13 and step 14 added) RTSS-3-1-25 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP, IPv6) (step 14 and step 15 added) RTSS-3-1-26 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) (step 16 and step 17 added) RTSS-3-1-27 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP, IPv6) (step 14 and step 15 added) RTSS-3-2-20 MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv4) (step 14 and step 15 added) RTSS-3-2-21 MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv6) (step 14 and step 15 added) RTSS-6-1-7 BACKCHANNEL – AAC (RTP-Unicast/UDP, IPv4) (step 11 added) RTSS-6-1-8 BACKCHANNEL – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4) (step 13 added) RTSS-6-1-9 BACKCHANNEL – AAC (RTP/RTSP/TCP, IPv4) (step 11 added) RTSS-6-1-12 BACKCHANNEL STREAMING BY POST – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4) (step 13 updated)
19.12	Aug 29, 2019	The following were changed according to #1888 and #1805: A.20 Media Profile Select or Creation for Audio and Video Streaming (removed) A.20 Media Profile Configuration with Required Audio Encoder and with Video Encoder (added)

		A.21 Media Profile Creation for Video Streaming (removed) A.21 Get Stream Uri (added) A.5 Media Profile Configuration for Audio Streaming (removed) A.5 Device Configuration for Audio Streaming with Media Profile (added) A.6 Media Profile Configuration for Audio & Video Streaming (removed) A.6 Device Configuration for Video and Audio Streaming (added) RTSS-5-1-8 START AND STOP MULTICAST STREAMING – G.711 (IPv4) (removed) The following test cases were updated to use GetCompatibleAudioSourceConfigurations instead of GetAudioSourceConfigurations: RTSS-2-1-28 AUDIO STREAMING – G.711 (RTP-Unicast/UDP) RTSS-2-1-29 AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-30 AUDIO STREAMING – G.711 (RTP/RTSP/TCP) RTSS-2-1-31 AUDIO STREAMING – G.726 (RTP-Unicast/UDP) RTSS-2-1-32 AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-33 AUDIO STREAMING – G.726 (RTP/RTSP/TCP) RTSS-2-1-34 AUDIO STREAMING – AAC (RTP-Unicast/UDP) RTSS-2-1-35 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-36 AUDIO STREAMING – AAC (RTP/RTSP/TCP) RTSS-2-1-46 AUDIO STREAMING – G.711 (RTP-Unicast/UDP, IPv6) RTSS-2-1-47 AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-48 AUDIO STREAMING – G.711 (RTP/RTSP/TCP, IPv6) RTSS-2-1-49 AUDIO STREAMING – G.726 (RTP-Unicast/UDP, IPv6) RTSS-2-1-50 AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-51 AUDIO STREAMING – G.726 (RTP/RTSP/TCP, IPv6) RTSS-2-1-52 AUDIO STREAMING – AAC (RTP-Unicast/UDP, IPv6) RTSS-2-1-53 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-54 AUDIO STREAMING – AAC (RTP/RTSP/TCP, IPv6)

		RTSS-2-2-11 MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv4) RTSS-2-2-12 MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv6) RTSS-2-2-13 MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv4) RTSS-2-2-14 MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv6) RTSS-2-2-15 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv4) RTSS-2-2-16 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv6) RTSS-5-1-10 START AND STOP MULTICAST STREAMING – G.711 (IPv4, ONLY AUDIO PROFILE) RTSS-5-1-11 START AND STOP MULTICAST STREAMING – G.726 (IPv4, ONLY AUDIO PROFILE) RTSS-5-1-12 START AND STOP MULTICAST STREAMING – AAC (IPv4, ONLY AUDIO PROFILE) The following test cases were updated with change of test name to configure media profile with any video encoder instead of JPEG only: RTSS-2-1-19 MEDIA STREAMING – G.711 (RTP-Unicast/UDP) RTSS-2-1-20 MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-21 MEDIA STREAMING – G.711 (RTP/RTSP/TCP) RTSS-2-1-22 MEDIA STREAMING – G.726 (RTP-Unicast/UDP) RTSS-2-1-23 MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-24 MEDIA STREAMING – G.726 (RTP/RTSP/TCP) RTSS-2-1-25 MEDIA STREAMING – AAC (RTP-Unicast/UDP) RTSS-2-1-26 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-27 MEDIA STREAMING – AAC (RTP/RTSP/TCP) RTSS-2-1-37 MEDIA STREAMING – G.711 (RTP-Unicast/UDP, IPv6) RTSS-2-1-38 MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-39 MEDIA STREAMING – G.711 (RTP/RTSP/TCP, IPv6) RTSS-2-1-40 MEDIA STREAMING – G.726 (RTP-Unicast/UDP, IPv6) RTSS-2-1-41 MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

		RTSS-2-1-42 MEDIA STREAMING – G.726 (RTP/RTSP/TCP, IPv6) RTSS-2-1-43 MEDIA STREAMING – AAC (RTP-Unicast/UDP, IPv6) RTSS-2-1-44 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-45 MEDIA STREAMING – AAC (RTP/RTSP/TCP, IPv6) RTSS-3-1-10 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/UDP) RTSS-3-1-11 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-3-1-12 MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP) RTSS-3-1-13 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/UDP) RTSS-3-1-14 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-3-1-15 MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP) RTSS-3-1-16 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP) RTSS-3-1-17 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP) RTSS-3-1-18 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP) RTSS-3-1-19 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/UDP, IPv6) RTSS-3-1-20 MEDIA STREAMING – JPEG/G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-3-1-21 MEDIA STREAMING – JPEG/G.711 (RTP/RTSP/TCP, IPv6) RTSS-3-1-22 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/UDP, IPv6) RTSS-3-1-23 MEDIA STREAMING – JPEG/G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-3-1-24 MEDIA STREAMING – JPEG/G.726 (RTP/RTSP/TCP, IPv6) RTSS-3-1-25 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/UDP, IPv6) RTSS-3-1-26 MEDIA STREAMING – JPEG/AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-3-1-27 MEDIA STREAMING – JPEG/AAC (RTP/RTSP/TCP, IPv6) RTSS-3-2-16 MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv4)

		RTSS-3-2-17 MEDIA STREAMING – JPEG/G.711 (RTP-Multicast/UDP, IPv6) RTSS-3-2-18 MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv4) RTSS-3-2-19 MEDIA STREAMING – JPEG/G.726 (RTP-Multicast/UDP, IPv6) RTSS-3-2-20 MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv4) RTSS-3-2-21 MEDIA STREAMING – JPEG/AAC (RTP-Multicast/UDP, IPv6) RTSS-3-2-22 VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT PORTS RTSS-3-2-23 VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT ADDRESS RTSS-5-1-9 START AND STOP MULTICAST STREAMING – JPEG/G.711 (IPv4)
19.12	Aug 29, 2019	The following were changed according to #1783: Normative references: link to RFC 2326 added, link to IETF RFC 4566 (added) New annexes added: A.30 Check IP Type of URI A.31 Check of IP address type in response to RTSP DESCRIBE The following annexes were updated: A.15 Start Media Stream Using RTP-Unicast/UDP Transport (step 2 and step 7 added) A.16 Start Media Stream Using RTP-Multicast/UDP Transport (step 2 and step 8 added) A.17 Start Media Stream Using RTP-Unicast/RTSP/HTTP/TCP Transport (step 2 and step 9 added) A.18 Start Media Stream Using RTP/RTSP/TCP Transport (step 2 and step 7 added) Notes with link to A.30 Check IP Type of URI and A.31 Check of IP address type in response to RTSP DESCRIBE added into the following test cases: RTSS-1-1-24 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION (ALL RESOLUTIONS) RTSS-1-1-25 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION (ALL RESOLUTIONS) RTSS-1-1-26 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION (ALL RESOLUTIONS) RTSS-1-1-27 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP)

		RTSS-1-1-28 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-29 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) RTSS-1-1-31 MEDIA CONTROL – RTSP/TCP RTSS-1-1-32 MEDIA STREAMING – RTSP KEEPALIVE (SET_PARAMETER) RTSS-1-1-33 MEDIA STREAMING - RTSP KEEPALIVE (OPTIONS) RTSS-1-1-34 MEDIA STREAMING – JPEG (RTP-Unicast / UDP) RTSS-1-1-35 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-36 MEDIA STREAMING – JPEG (RTP/RTSP/TCP) RTSS-1-1-37 MEDIA STREAMING – MPEG4 (RTP-Unicast/ UDP) RTSS-1-1-38 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-39 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP) RTSS-1-1-40 SET SYNCHRONIZATION POINT – MPEG4 RTSS-1-1-41 MEDIA STREAMING – H.264 (RTP-Unicast/ UDP) RTSS-1-1-42 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-1-1-43 MEDIA STREAMING – H.264 (RTP/RTSP/TCP) RTSS-1-1-44 SET SYNCHRONIZATION POINT – H.264 RTSS-1-1-45 MEDIA STREAMING – RTP-Unicast/RTSP/HTTP/TCP (LINE BREAKS IN BASE64 ENCODING) RTSS-1-1-46 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION RTSS-1-1-47 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION RTSS-1-1-48 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION RTSS-1-1-49 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-1-1-50 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-1-1-51 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-1-1-53 MEDIA STREAMING – JPEG (VALIDATING RTP HEADER EXTENSION)

		RTSS-1-1-54 MEDIA STREAMING – JPEG (RTP-Unicast/UDP, IPv6) RTSS-1-1-55 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-1-1-56 MEDIA STREAMING – JPEG (RTP/RTSP/TCP, IPv6) RTSS-1-1-57 MEDIA STREAMING – MPEG4 (RTP-Unicast/UDP, IPv6) RTSS-1-1-58 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-1-1-59 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP, IPv6) RTSS-1-1-60 MEDIA STREAMING – H.264 (RTP-Unicast/UDP, IPv6) RTSS-1-1-61 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-1-1-62 MEDIA STREAMING – H.264 (RTP/RTSP/TCP, IPv6) RTSS-1-2-12 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) RTSS-1-2-13 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4) RTSS-1-2-14 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4) RTSS-1-2-15 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4) RTSS-1-2-16 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv6) RTSS-1-2-17 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv6) RTSS-1-2-18 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv6) RTSS-1-2-21 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv4) RTSS-1-2-22 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv6) RTSS-1-2-23 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS) RTSS-2-1-28 AUDIO STREAMING – G.711 (RTP-Unicast/UDP) RTSS-2-1-29 AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-30 AUDIO STREAMING – G.711 (RTP/RTSP/TCP) RTSS-2-1-31 AUDIO STREAMING – G.726 (RTP-Unicast/UDP)

		RTSS-2-1-32 AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-33 AUDIO STREAMING – G.726 (RTP/RTSP/TCP) RTSS-2-1-34 AUDIO STREAMING – AAC (RTP-Unicast/UDP) RTSS-2-1-35 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP) RTSS-2-1-36 AUDIO STREAMING – AAC (RTP/RTSP/TCP) RTSS-2-1-46 AUDIO STREAMING – G.711 (RTP-Unicast/UDP, IPv6) RTSS-2-1-47 AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-48 AUDIO STREAMING – G.711 (RTP/RTSP/TCP, IPv6) RTSS-2-1-49 AUDIO STREAMING – G.726 (RTP-Unicast/UDP, IPv6) RTSS-2-1-50 AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-51 AUDIO STREAMING – G.726 (RTP/RTSP/TCP, IPv6) RTSS-2-1-52 AUDIO STREAMING – AAC (RTP-Unicast/UDP, IPv6) RTSS-2-1-53 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6) RTSS-2-1-54 AUDIO STREAMING – AAC (RTP/RTSP/TCP, IPv6) RTSS-2-2-11 MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv4) RTSS-2-2-12 MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv6) RTSS-2-2-13 MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv4) RTSS-2-2-14 MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv6) RTSS-2-2-15 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv4) RTSS-2-2-16 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv6) RTSS-4-1-3 NOTIFICATION STREAMING RTSS-6-1-1 BACKCHANNEL – G.711 (RTP-Unicast/UDP, IPv4) RTSS-6-1-2 BACKCHANNEL – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-3 BACKCHANNEL – G.711 (RTP/RTSP/TCP, IPv4) RTSS-6-1-4 BACKCHANNEL – G.726 (RTP-Unicast/UDP, IPv4)

		RTSS-6-1-5 BACKCHANNEL – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-6 BACKCHANNEL – G.726 (RTP/RTSP/TCP, IPv4) RTSS-6-1-7 BACKCHANNEL – AAC (RTP-Unicast/UDP, IPv4) RTSS-6-1-8 BACKCHANNEL – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-9 BACKCHANNEL – AAC (RTP/RTSP/TCP, IPv4) RTSS-6-1-10 BACKCHANNEL STREAMING BY POST – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-11 BACKCHANNEL STREAMING BY POST – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-12 BACKCHANNEL STREAMING BY POST – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4)
19.12	Sep 24, 2019	The following annex was added according to #1808: A.29 Removing Configurations from Media Profile (steps 4-5 added) RTSS-5-1-7 START AND STOP MULTICAST STREAMING – JPEG (IPv4) (step 6 updated with adding of AudioSourceConfiguration in the list of configuration types) RTSS-1-2-19 VIDEO ENCODER CONFIGURATION – MULTICAST PORT (IPv4) (step 11 updated with adding of AudioSourceConfiguration in the list of configuration types) RTSS-1-2-20 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4) (step 11 updated with adding of AudioSourceConfiguration in the list of configuration types)
20.06	Jan 31, 2020	The following annex was updated according to #1965: Annex A.28 Video Encoder Configuration – Frame Rate Selection (framerate selection logic for the highest JPEG resolution was updated)
21.06	Apr 09, 2021	The following annexes were updated according to #2124: A.4 Turn on IPv6 network interface A.5 Restore Network Settings
21.06	May 26, 2021	The following test cases were removed according to #2215: RTSS-6-1-2 BACKCHANNEL – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-5 BACKCHANNEL – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv4) RTSS-6-1-8 BACKCHANNEL – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4)
23.06	Jun 20, 2023	Mentioning of incorrect GetStreamUriResponse parameters were removed according to #121.
24.12	Dec 02, 2024	Requirement about DUT configuration with 0 rotation was added to the Pre-Requisite of all test cases involving video streaming in the scope of #270 ticket.

Table of Contents

1 Introduction

The goal of the ONVIF test specification set is to make it possible to realize fully interoperable IP physical security implementation from different vendors. The set of ONVIF test specification describes the test cases need to verify the [ONVIF Network Interface Specs] and [ONVIF Conformance] requirements. And also 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 Real Time Streaming Device Test Specification acts as a supplementary document to the [ONVIF Network Interface Specs], illustrating test cases need to be executed and passed. And also this specification acts as an input document to the development of test tool which 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 Real Time Streaming Device Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant devices. Conformance testing is meant to be functional black-box testing. The objective of this specification to provide test cases to test individual requirements of ONVIF devices according to ONVIF Media Service and the Realtime Streaming Specification which is defined in [ONVIF Network Interface Specs].

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for [ONVIF Network Interface Specs].

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

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

  Wi-Fi Conformance test

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

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

1.2 Real Time Streaming

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

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

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

2 Normative references

- [ONVIF Conformance] ONVIF Conformance Process Specification:

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

- [ONVIF Profile Policy] ONVIF Profile Policy:

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

- [ONVIF Network Interface Specs] ONVIF Network Interface Specification documents:

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

- [ONVIF Core Specs] ONVIF Core Specification:

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

- [ONVIF Media Spec] ONVIF Media Service Specification:

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

- [ONVIF Streaming Spec] ONVIF Streaming Specification:

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

- [ONVIF Base Test] ONVIF Base Device Test Specification:

  https://www.onvif.org/profiles/conformance/device-test/

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

  http://www.iso.org/directives

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

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

- [SOAP 1.2, Part 1] W3C SOAP 1.2, Part 1, Messaging Framework:

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

- [XML-Schema, Part 1] W3C XML Schema Part 1: Structures Second Edition:

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

- [XML-Schema, Part 2] W3C XML Schema Part 2: Datatypes Second Edition:

  http://www.w3.org/TR/xmlschema-2/

- [WS-Security] "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard, February 2006.:

  http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

- IETF RFC 2326, Real Time Streaming Protocol (RTSP)

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

3 Terms and Definitions

3.1 Conventions

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

3.2 Definitions

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

Profile See ONVIF Profile Policy.

ONVIF Device Computer appliance or software program that exposes one or multiple ONVIF Web Services.

ONVIF Client Computer appliance or software program that uses ONVIF Web Services.

Configuration Entity A network video device media abstract component that is used to produce a media stream on the network, i.e. video and/or audio stream.

Media Profile A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations.

SOAP SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.

Device Test Tool ONVIF Device Test Tool that tests ONVIF Device implementation towards the ONVIF Test Specification set.

Media Service Services to determine the streaming properties of requested media streams.

3.3 Abbreviations

This section describes abbreviations used in this document.

HTTP	Hyper Text Transport Protocol.
AAC	Advanced Audio Coding.
URI	Uniform Resource Identifier.
WSDL	Web Services Description Language.
XML	eXtensible Markup Language.
TTL	Time To Live.

JPEG	Joint Photographic Experts Group.
MPEG-4	Moving Pictures Experts Group-4.
QVGA	Quarter Video Graphics Array.

4 Test Overview

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

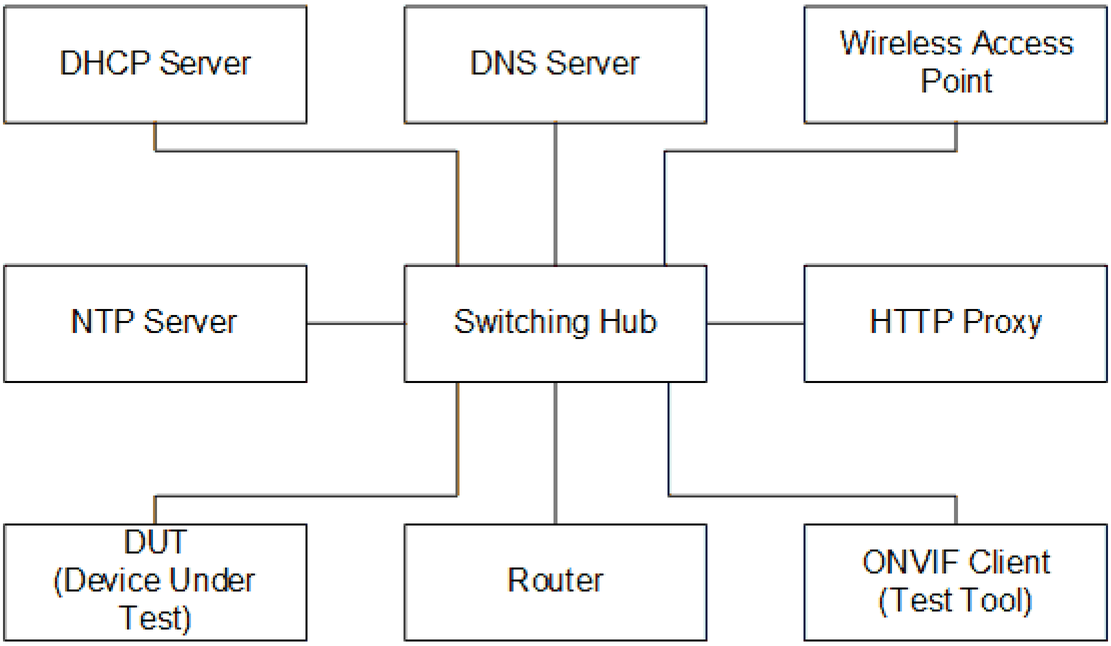
4.1 Test Setup

4.1.1 Network Configuration for DUT

The generic test configuration for the execution of test cases defined in this document is as shown below (Figure 4.1).

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 4.1. 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 behavior of the DUT. It handles both expected and unexpected behavior.

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

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

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

DHCP Server: provides IPv4 Address to the connected devices.

NTP Server: provides time synchronization between ONVIF Client and DUT.

Switching Hub: provides network connectivity among all the test equipments in the test environment. All devices should be connected to the Switching Hub. When running multiple test instances in parallel on the same network, the Switching Hub should be configured to use filtering in order to avoid multicast traffic being flooded to all ports, because this may affect test stability.

Router: provides router advertisements for IPv6 configuration.

4.2 Prerequisites

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

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

4.3 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.

4.3.1 Real Time Streaming

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

ONVIF Client shall explicitly specify the optional transport protocols supported by DUT.

ONVIF Client and DUT time should be synchronized for media streaming.

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

Poor streaming test is outside the scope of the ONVIF Test Specification

Please refer to Annex A.2 for the correct interpretation of StreamSetup syntax.

Please refer to Section 5 for Real Time Streaming Test Cases.

5 Real Time Streaming Test Cases

5.1 Video Streaming

5.1.1 Unicast

5.1.1.1 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION (ALL RESOLUTIONS)

Test Case ID: RTSS-1-1-24

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri, Live Streaming.

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of all different Resolutions for video encoder configuration and video stream.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.

4. Verify the **GetVideoEncoderConfigurationsResponse** message.

5. Find or create media profile with Video Source Configuration and Video Encoder Configuration with token VECToken1 and supporting of JPEG encoding, where VECToken1 is first video encoder configuration token from

GetVideoEncoderConfigurationsResponse message by following the procedure mentioned in Annex A.13. If it is not possible skip steps 6-26 and go to the step 27.

6. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “Profile1”, where “Profile1” is profile token from the step 5) to get video encoder configuration options.

7. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.

8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = JPEG, where [Width1, Height1] is the first resolution from the Options.JPEG.ResolutionsAvailable) to change video encoder configuration.

9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

10. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

11. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = JPEG) from the DUT. Check that new setting for Resolution and Encoding was applied.

12. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

13. DUT sends RTSP URI in the **GetStreamUriResponse** message.

14. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

15. ONVIF Client invokes **RTSP DESCRIBE** request.

16. DUT sends **200 OK** message and SDP information.

17. ONVIF Client invokes **RTSP SETUP** request with transport parameter as RTP/UDP.

18. DUT sends **200 OK** message and the media stream information.

19. ONVIF Client invokes **RTSP PLAY** request.

20. DUT sends **200 OK** message and starts media streaming.

21. DUT sends **JPEG RTP** media stream to ONVIF Client over UDP. Verify that stream has JPEG encoding and [Width1, Height1] resolution. (Note: if stream Height resolution is between Height1 – 8 and Height1 + 8 it is assumed as valid; if stream Width resolution is between Width1 – 8 and Width1 + 8 it is assumed as valid.)

22. DUT sends RTCP sender report to ONVIF Client.

23. DUT validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client invokes **RTSP TEARDOWN** control request at the end of media streaming to terminate the RTSP session.

25. DUT sends **200 OK** Response and terminates the RTSP Session.

26. Repeat 8-25 steps for the rest JPEG Resolutions supported by selected configuration.

27. Repeat steps 5-26 for the rest Video Encoder configurations supported by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP **200 OK** response for **RTSP DESCRIBE**, **SETUP**, **PLAY** and **TEARDOWN** requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not sent valid video stream with JPEG encoding and specified resolution or sent it in another encoding or resolution.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.28 for Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.2 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION (ALL RESOLUTIONS)

Test Case ID: RTSS-1-1-25

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri, Live Streaming.

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of all different Resolutions for video encoder configuration and video stream.

Pre-Requisite: Media is supported by DUT. MPEG4 is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.

4. Verify the **GetVideoEncoderConfigurationsResponse** message.

5. Find or create media profile with Video Source Configuration and Video Encoder Configuration with token VECToken1 and supporting of MPEG4 encoding, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message by following the procedure mentioned in Annex A.13. If it is not possible skip steps 6-26 and go to the step 27.

6. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “Profile1”, where “Profile1” is profile token from the step 5) to get video encoder configuration options.

7. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.

8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = MPEG4, where [Width1, Height1] is the first resolution from the Options.MPEG4.ResolutionsAvailable) to change video encoder configuration.

9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

10. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

11. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = MPEG4) from the DUT. Check that new setting for Resolution and Encoding was applied.

12. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

13. DUT sends RTSP URI in the **GetStreamUriResponse** message.

14. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

15. ONVIF Client invokes RTSP DESCRIBE request.

16. DUT sends 200 OK message and SDP information.

17. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

18. DUT sends 200 OK message and the media stream information.

19. ONVIF Client invokes RTSP PLAY request.

20. DUT sends 200 OK message and starts media streaming.

21. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP. Verify that stream has MPEG4 encoding and [Width1, Height1] resolution. (Note: if stream Height resolution is between Height1 – 8 and Height1 + 8 it is assumed as valid; if stream Width resolution is between Width1 – 8 and Width1 + 8 it is assumed as valid.)

22. DUT sends RTCP sender report to ONVIF Client.

23. DUT validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

25. DUT sends 200 OK Response and terminates the RTSP Session.

26. Repeat 8-25 steps for the rest MPEG4 Resolutions supported by selected configuration.

27. Repeat steps 5-26 for the rest Video Encoder configurations supported by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

- DUT did not sent valid video stream with MPEG4 encoding and specified resolution or sent it in another encoding or resolution.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.28 for Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.3 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION (ALL RESOLUTIONS)

Test Case ID: RTSS-1-1-26

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri, Live Streaming.

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of all different Resolutions for video encoder configuration and video stream.

Pre-Requisite: Media is supported by DUT. H.264 is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.

4. Verify the **GetVideoEncoderConfigurationsResponse** message.

5. Find or create media profile with Video Source Configuration and Video Encoder Configuration with token VECToken1 and supporting of H.264 encoding, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message by following the procedure mentioned in Annex A.13. If it is not possible skip steps 6-26 and go to the step 27.

6. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “Profile1”, where “Profile1” is profile token from the step 5) to get video encoder configuration options.

7. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.

8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = H.264, where [Width1, Height1] is the first resolution from the Options.H.264.ResolutionsAvailable) to change video encoder configuration.

9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

10. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

11. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = H.264) from the DUT. Check that new setting for Resolution and Encoding was applied.

12. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

13. DUT sends RTSP URI in the **GetStreamUriResponse** message.

14. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

15. ONVIF Client invokes RTSP DESCRIBE request.

16. DUT sends 200 OK message and SDP information.

17. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

18. DUT sends 200 OK message and the media stream information.

19. ONVIF Client invokes RTSP PLAY request.

20. DUT sends 200 OK message and starts media streaming.

21. DUT sends H.264 RTP media stream to ONVIF Client over UDP. Verify that stream has H.264 encoding and [Width1, Height1] resolution. (Note: if stream Height resolution is between Height1 – 8 and Height1 + 8 it is assumed as valid; if stream Width resolution is between Width1 – 8 and Width1 + 8 it is assumed as valid.)

22. DUT sends RTCP sender report to ONVIF Client.

23. DUT validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

25. DUT sends 200 OK Response and terminates the RTSP Session.

26. Repeat 8-25 steps for the rest H.264 Resolutions supported by selected configuration.

27. Repeat steps 5-26 for the rest Video Encoder configurations supported by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not sent valid video stream with H.264 encoding and specified resolution or sent it in another encoding or resolution.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.28 for Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.4 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP)

Test Case ID: RTSS-1-1-27

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances.

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number of Video Encoder Instances (RTP-Unicast/UDP).

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

9. DUT sends RTSP URI in the **GetStreamUriResponse** message.

10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

11. ONVIF Client invokes RTSP DESCRIBE request.

12. DUT sends 200 OK message and SDP information.

13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

14. DUT sends 200 OK message and the media stream information.

15. ONVIF Client invokes RTSP PLAY request.

16. DUT sends 200 OK message and starts media streaming.

17. DUT sends JPEG RTP media stream to ONVIF Client over UDP.

18. DUT sends RTCP sender report to ONVIF Client.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. Repeat steps 8-20 to start video streaming for all profiles from step 7.

21. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

22. DUT sends 200 OK Response and terminates the RTSP Session.

23. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than one, repeat steps 5-22 for the last video source configuration token from **GetVideoEncoderConfigurationsResponse** message.

24. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than two, repeat steps 5-22 for the video source configuration token between the first and the last VSC tokens from **GetVideoSourceConfigurationsResponse** message.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

- DUT did not send valid video stream.

- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.5 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-1-1-28

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances.

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP-Unicast/RTSP/HTTP/TCP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.
4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the first media profile from step 7.

9. DUT sends RTSP URI in the **GetStreamUriResponse** message.

10. ONVIF Client verifies the HTTP media stream URI provided by the DUT.

11. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

12. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

13. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

14. DUT sends 200 OK message and SDP information on HTTP GET connection.

15. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter for both audio and video streams separately.

16. DUT sends 200 OK message and the media stream information on HTTP GET connection.

17. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

18. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

19. DUT transfers RTP media stream to ONVIF Client on HTTP GET connection.

20. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

21. DUT validates the received RTP and RTCP packets, decodes and renders them.

22. Repeat steps 8-22 to start video streaming for all profiles from step 7.

23. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

24. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

25. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than one, repeat steps 5-24 for the last video source configuration token from **GetVideoEncoderConfigurationsResponse** message.

26. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than two, repeat steps 5-24 for the video source configuration token between the first and the last VSC tokens from **GetVideoSourceConfigurationsResponse** message.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.6 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP)

Test Case ID: RTSS-1-1-29

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances.

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP/RTSP/TCP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. RTP/RTSP/TCP media streaming is implemented by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the first media profile from step 7.

9. DUT sends RTSP URI in the **GetStreamUriResponse** message.

10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

11. ONVIF Client invokes RTSP DESCRIBE request.

12. DUT sends 200 OK message and SDP information.

13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/TCP.

14. DUT sends 200 OK message and the media stream information.

15. ONVIF Client invokes RTSP PLAY request.

16. DUT sends 200 OK message and starts media streaming.

17. DUT sends RTP media stream to ONVIF Client over TCP.

18. DUT sends RTCP sender report to ONVIF Client.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. Repeat steps 8-20 to start video streaming for all profiles from step 7.

21. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

22. DUT sends 200 OK Response and terminates the RTSP Session.

23. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than one, repeat steps 5-22 for the last video source configuration token from **GetVideoEncoderConfigurationsResponse** message.

24. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than two, repeat steps 5-22 for the video source configuration token between the first and the last VSC tokens from **GetVideoSourceConfigurationsResponse** message.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.7 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES)

Test Case ID: RTSS-1-1-30

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances.

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (Mix Of Transport Types)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. RTP/RTSP/TCP media streaming is implemented by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. For each Media Profile (*profile*) that will be used for multicast streaming:

   8.1. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

   - in *profile* - Media Profile to update

9. ONVIF Client start media streaming for first profile to retrieve media stream URI for the first media profile from step 7. (To start media streaming Annex A.15, Annex A.16, Annex A.17, or Annex A.18 will be used in loop. Annex A.16 will be used if RTP-Multicast/UDP supported. Annex A.18 will be used if RTP/RTSP/TCP supported.)

10. Repeat step 8 to start video streaming for all profiles from step 7 (transport type should be altered for each iteration of the loop, see step 8 for more details).

11. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

12. DUT sends 200 OK Response and terminates the RTSP Session.

13. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than one, repeat steps 5-11 for the last video source configuration token from **GetVideoEncoderConfigurationsResponse** message.

14. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than two, repeat steps 5-11 for the video source configuration token between the first and the last VSC tokens from **GetVideoSourceConfigurationsResponse** message.

15. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: In case if there are more than one Media Profiles with the same Video Encoder Configuration, multicast streaming will be started only for one of them.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.8 MEDIA CONTROL – RTSP/TCP

Test Case ID: RTSS-1-1-31

Specification Coverage: Stream control, RTSP.

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify RTSP control messages of DUT.

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI in the **GetStreamUriResponse** message.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client will invoke RTSP OPTIONS control request to understand the RTSP methods supported by DUT.

10. DUT sends 200 OK Response and list of supported RTSP methods.

11. ONVIF Client will invoke RTSP DESCRIBE control request to retrieve the media description information.

12. DUT sends 200 OK Response and SDP message.

13. ONVIF Client validates the session description information in the SDP message.

14. ONVIF Client will invoke RTSP SETUP control request to create a RTSP Session.

15. DUT sends 200 OK Response and Stream Information details.

16. ONVIF Client Verifies “Transport”, “Session” and “timeout” header fields in the SETUP response message.

17. ONVIF Client will invoke RTSP PLAY control request to initiate the media streaming.

18. DUT sends 200 OK Response and RTP protocol information.

19. ONVIF Client verifies “Session”, “RTP-Info”, “seq”, “uri” and “rtptime” header fields in the PLAY response message.

20. DUT transfers media streams over RTP/UDP.

21. DUT sends RTCP sender report to ONVIF Client.

22. ONVIF Client validates RTCP packets.

23. ONVIF Client validates RTP header for each media stream and render it after the validation.

24. ONVIF Client will invoke RTSP TEARDOWN control request to terminate the RTSP session at the end of the streaming.

25. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send correct media stream information in the SDP message.
- DUT did not send mandatory headers or fields in the SETUP response message.
- DUT did not send mandatory headers or fields in the PLAY response message.
- DUT did not send RTSP 200 OK response for RTSP OPTIONS, DESCRIBE, SETUP and PLAY requests.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.

Note: If DUT does not send SET_PARAMETER in OPTIONS response, the Device Test Tool provides warning.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.9 MEDIA STREAMING – RTSP KEEPALIVE (SET_PARAMETER)

Test Case ID: RTSS-1-1-32

Specification Coverage: Keep-alive method for RTSP session.

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify ONVIF Client and DUT exchange SET_PARAMETER messages during an active streaming session.

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

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

7. DUT sends RTSP URI in the **GetStreamUriResponse** message.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client will invoke RTSP control requests (DESCRIBE, SETUP and PLAY).

10. ONVIF Client will verify “Timeout” header in the SETUP Response from DUT.

11. Based on the “Timeout” value, ONVIF Client will invoke RTSP SET_PARAMETER messages.

12. DUT will respond with 200 OK for RTSP SET_PARAMETER request.

13. Verify that the ONVIF Client and DUT are exchanging periodic SET_PARAMETER messages while a stream is being delivered.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

- DUT did not send Timeout header in RTSP SETUP RESPONSE.

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.10 MEDIA STREAMING - RTSP KEEPALIVE (OPTIONS)

Test Case ID: RTSS-1-1-33

Specification Coverage: Keep-alive method for RTSP session.

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify ONVIF Client and DUT exchange OPTIONS messages during an active streaming session.

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

7. Verify the **GetStreamUriResponse** message from the DUT.

8. Verify the RTSP media stream URI provided by the DUT.

9. ONVIF Client will invoke RTSP control requests (DESCRIBE, SETUP and PLAY).

10. ONVIF Client will verify “Timeout” header in the SETUP Response from DUT.

11. Based on the “Timeout” value, ONVIF Client will invoke RTSP OPTIONS messages.

12. DUT will respond with 200 OK for RTSP OPTIONS request.

13. Verify that the ONVIF Client and DUT are exchanging periodic OPTIONS messages while a stream is being delivered.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send Timeout header in RTSP SETUP RESPONSE.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and OPTIONS requests.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.11 MEDIA STREAMING – JPEG (RTP-Unicast/UDP)

Test Case ID: RTSS-1-1-34

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

Command Under Test: None.

WSDL Reference: media.wsdl

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

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

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

15. DUT sends JPEG RTP media stream to ONVIF Client over UDP.

16. DUT sends RTCP sender report to ONVIF Client.

17. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

19. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

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

Test Case ID: RTSS-1-1-35

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

Command Under Test: None.

WSDL Reference: media.wsdl

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

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.
4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.
5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends HTTP URI in the **GetStreamUriResponse** message.
8. ONVIF Client verifies the HTTP media stream URI provided by the DUT.
9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

11. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

12. DUT sends 200 OK message and SDP information on HTTP GET connection.

13. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

14. DUT sends 200 OK message and the media stream information on HTTP GET connection.

15. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

16. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

17. DUT transfers JPEG RTP media stream to ONVIF Client on HTTP GET connection.

18. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

21. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.13 MEDIA STREAMING – JPEG (RTP/RTSP/TCP)

Test Case ID: RTSS-1-1-36

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

Command Under Test: None.

WSDL Reference: media.wsdl

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

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

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

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI in the **GetStreamUriResponse** message.

8. Verify the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request.

10. DUT sends 200 OK message and SDP information.

11. ONVIF Client invokes RTSP SETUP request with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

12. DUT sends 200 OK message and the media stream information.

13. ONVIF Client invokes RTSP PLAY request.

14. DUT sends 200 OK message and starts media streaming.

15. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.

16. 16. DUT validates the received RTP and RTCP packets, decodes and renders them.

17. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

18. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.14 MEDIA STREAMING – MPEG4 (RTP-Unicast/UDP)

Test Case ID: RTSS-1-1-37

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on RTP/UDP Unicast Transport.

Pre-Requisite: MPEG4 is implemented by DUT. A media profile with MPEG4 video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence =

false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI in the **GetStreamUriResponse** message.
8. Verify the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.
12. DUT sends 200 OK message and the media stream information.
13. ONVIF Client invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP.
16. DUT sends RTCP sender report to ONVIF Client.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.

- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.15 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-1-1-38

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on HTTP Transport.

Pre-Requisite: MPEG4 is implemented by DUT. A media profile with MPEG4 video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends HTTP URI in the **GetStreamUriResponse** message.

8. ONVIF Client verifies the HTTP media stream URI provided by the DUT.

9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

11. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

12. DUT sends 200 OK message and SDP information on HTTP GET connection.

13. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

14. DUT sends 200 OK message and the media stream information on HTTP GET connection.

15. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

16. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

17. DUT transfers MPEG4 RTP media stream to ONVIF Client on HTTP GET connection.

18. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

21. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.16 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP)

Test Case ID: RTSS-1-1-39

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on RTP/RTSP/TCP using RTSP tunnel.

Pre-Requisite: MPEG4 and RTP/RTSP/TCP media streaming is implemented by DUT. A media profile with MPEG4 video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.
4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.
5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI in the **GetStreamUriResponse** message.
8. Verify the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. ONVIF Client invokes RTSP SETUP request with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.
12. DUT sends 200 OK message and the media stream information.
13. ONVIF Client invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.

16. DUT validates the received RTP and RTCP packets, decodes and renders them.
17. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
18. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.17 SET SYNCHRONIZATION POINT – MPEG4

Test Case ID: RTSS-1-1-40

Specification Coverage: Set synchronization point.

Command Under Test: SetSynchronizationPoint

WSDL Reference: media.wsdl

Test Purpose: To request synchronization point from DUT for MPEG4 media stream.

Pre-Requisite: MPEG4 is implemented by DUT. Real-time streaming supported by DUT. A media profile with MPEG4 video encoder configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.
4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.
5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI in the **GetStreamUriResponse** message.
8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.
12. DUT sends 200 OK message and the media stream information.
13. ONVIF Client invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP.

16. DUT sends RTCP sender report to ONVIF Client.

17. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. ONVIF Client invokes SetSynchronizationPoint request on the selected media profile.

19. DUT sends the SetSynchronizationPoint response indicating success.

20. DUT inserts the I-frame in the ongoing media stream.

21. ONVIF Client verifies that I-frame is sent by DUT before the regular 'I-frame insertion time interval'.

22. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

23. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

- DUT did not send **SetSynchronizationPointResponse** message.

- DUT did not send I-frame before the regular 'I-frame insertion time interval' upon invoking SetSynchronizationPoint request.

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.18 MEDIA STREAMING – H.264 (RTP-Unicast/UDP)

Test Case ID: RTSS-1-1-41

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on RTP/UDP Unicast Transport.

Pre-Requisite: H.264 is implemented by DUT. Real-time streaming supported by DUT. A media profile with H.264 video encoder configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI in the **GetStreamUriResponse** message.

8. Verify the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.
12. DUT sends 200 OK message and the media stream information.
13. ONVIF Client invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT sends H.264 RTP media stream to ONVIF Client over UDP.
16. DUT sends RTCP sender report to ONVIF Client.
17. DUT validates the received RTP and RTCP packets, decodes and renders them.
18. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
19. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.19 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-1-1-42

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on HTTP Transport.

Pre-Requisite: H.264 is implemented by DUT. A media profile with H.264 video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends HTTP URI in the **GetStreamUriResponse** message.

8. ONVIF Client verifies the HTTP media stream URI provided by the DUT.

9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

11. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

12. DUT sends 200 OK message and SDP information on HTTP GET connection.

13. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

14. DUT sends 200 OK message and the media stream information on HTTP GET connection.

15. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

16. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

17. DUT transfers H.264 RTP media stream to ONVIF Client on HTTP GET connection.

18. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

21. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.20 MEDIA STREAMING – H.264 (RTP/RTSP/TCP)

Test Case ID: RTSS-1-1-43

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on RTP/RTSP/TCP using RTSP tunnel.

Pre-Requisite: H.264 and RTP/RTSP/TCP media streaming is implemented by DUT. A media profile with H.264 video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.
4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.
5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends RTSP URI in the **GetStreamUriResponse** message.
8. Verify the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request.
10. DUT sends 200 OK message and SDP information.
11. ONVIF Client invokes RTSP SETUP request with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.
12. DUT sends 200 OK message and the media stream information.
13. ONVIF Client invokes RTSP PLAY request.
14. DUT sends 200 OK message and starts media streaming.
15. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.
16. DUT validates the received RTP and RTCP packets, decodes and renders them.
17. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
18. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.21 SET SYNCHRONIZATION POINT – H.264

Test Case ID: RTSS-1-1-44

Specification Coverage: Set synchronization point.

Command Under Test: SetSynchronizationPoint

WSDL Reference: media.wsdl

Test Purpose: To request synchronization point from DUT for H.264 media stream.

Pre-Requisite: H.264 is implemented by DUT. Real-time streaming supported by DUT. A media profile with H.264 video encoder configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the GetVideoEncoderConfigurationOptionsResponse in Annex A.4.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI in the **GetStreamUriResponse** message.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request.

10. DUT sends 200 OK message and SDP information.

11. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

12. DUT sends 200 OK message and the media stream information.

13. ONVIF Client invokes RTSP PLAY request.

14. DUT sends 200 OK message and starts media streaming.

15. DUT sends H.264 RTP media stream to ONVIF Client over UDP.

16. DUT sends RTCP sender report to ONVIF Client.

17. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. ONVIF Client invokes **SetSynchronizationPoint** request on the selected media profile.

19. DUT sends the **SetSynchronizationPointResponse** indicating success.

20. DUT inserts the I-frame in the ongoing media stream.

21. ONVIF Client verifies that I-frame is sent by DUT before the regular ‘I-frame insertion time interval’.

22. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

23. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

- DUT did not send **SetSynchronizationPointResponse** message.

- DUT did not send I-frame before the regular ‘I-frame insertion time interval’ upon invoking **SetSynchronizationPoint** request.

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.22 MEDIA STREAMING – RTP-Unicast/RTSP/HTTP/TCP (LINE BREAKS IN BASE64 ENCODING)

Test Case ID: RTSS-1-1-45

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify JPEG media streaming based on HTTP Transport with line breaks for Base64 encoding in RTSP requests.

Pre-Requisite: A media profile with JPEG video encoder configuration. Real-time streaming supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.
4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.
5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.
7. DUT sends HTTP URI in the **GetStreamUriResponse** message.
8. ONVIF Client verifies the HTTP media stream URI provided by the DUT.
9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

11. ONVIF Client invokes RTSP DESCRIBE request with line breaks in Base64 encoding on HTTP POST connection.

12. DUT sends 200 OK message and SDP information on HTTP GET connection.

13. ONVIF Client invokes RTSP SETUP request with line breaks in Base64 encoding on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

14. DUT sends 200 OK message and the media stream information on HTTP GET connection.

15. ONVIF Client invokes RTSP PLAY request with line breaks in Base64 encoding on HTTP POST connection.

16. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

17. DUT transfers JPEG RTP media stream to ONVIF Client on HTTP GET connection.

18. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request with line breaks in Base64 encoding on HTTP POST connection and closes the HTTP POST connection.

21. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests with line breaks in Base64 encoding.
- DUT did not send valid RTP header in one or more media streams.

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: RTSP requests for steps 11, 13, 15, and 20 will be sent with line breaks in RTSP in Base64 encoding (see http://tools.ietf.org/html/rfc1421#section-4.3.2.4 for more details). These line breaks shall be ignored by the DUT during request processing.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.23 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION

Test Case ID: RTSS-1-1-46

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri, Live Streaming

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of different Resolutions for video encoder configuration and video stream.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. There is Video Encoder Configuration which is either configured with JPEG or the DUT allows to change encoding of Video Encoder Configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.
4. Verify the **GetVideoEncoderConfigurationsResponse** message.
5. Find or create media profile with Video Source Configuration and Video Encoder Configuration with token VECToken1 and supporting of JPEG encoding, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message by following the procedure mentioned in Annex A.13. If it is not possible skip steps 6-61 and go to the step 62.
6. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “Profile1”, where “Profile1” is profile token from the step 5) to get video encoder configuration options.
7. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.
8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = JPEG, where [Width1, Height1] is maximum resolution from the Options.JPEG.ResolutionsAvailable) to change video encoder configuration.
9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.
10. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.
11. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = JPEG) from the DUT. Check that new setting for Resolution and Encoding was applied.
12. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.
13. DUT sends RTSP URI in the **GetStreamUriResponse** message.
14. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
15. ONVIF Client invokes RTSP DESCRIBE request.

16. DUT sends 200 OK message and SDP information.

17. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

18. DUT sends 200 OK message and the media stream information.

19. ONVIF Client invokes RTSP PLAY request.

20. DUT sends 200 OK message and starts media streaming.

21. DUT sends JPEG RTP media stream to ONVIF Client over UDP. Verify that stream has JPEG encoding and [Width1, Height1] resolution. (Note: if stream Height resolution is between Height1 – 8 and Height1 + 8 it is assumed as valid; if stream Width resolution is between Width1 – 8 and Width1 + 8 it is assumed as valid.)

22. DUT sends RTCP sender report to ONVIF Client.

23. DUT validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

25. DUT sends 200 OK Response and terminates the RTSP Session.

26. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width2, Height2], Encoding = JPEG, where [Width2, Height2] is minimum resolution from the Options.JPEG.ResolutionsAvailable) to change video encoder configuration.

27. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

28. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

29. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width2, Height2], Encoding = JPEG, where [Width2, Height2]) from the DUT. Check that new setting for Resolution and Encoding was applied.

30. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

31. DUT sends RTSP URI in the **GetStreamUriResponse** message.

32. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

33. ONVIF Client invokes RTSP DESCRIBE request.

34. DUT sends 200 OK message and SDP information.

35. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

36. DUT sends 200 OK message and the media stream information.

37. ONVIF Client invokes RTSP PLAY request.

38. DUT sends 200 OK message and starts media streaming.

39. DUT sends JPEG RTP media stream to ONVIF Client over UDP. Verify that stream has JPEG encoding and [Width2, Height2] resolution.

40. DUT sends RTCP sender report to ONVIF Client.

41. DUT validates the received RTP and RTCP packets, decodes and renders them.

42. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

43. DUT sends 200 OK Response and terminates the RTSP Session.

44. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width3, Height3], Encoding = JPEG, where [Width3, Height3] is middle resolution from the Options.JPEG.ResolutionsAvailable) to change video encoder configuration.

45. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

46. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

47. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width3, Height3], Encoding = JPEG) from the DUT. Check that new setting for Resolution and Encoding was applied.

48. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

49. DUT sends RTSP URI in the **GetStreamUriResponse** message.

50. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

51. ONVIF Client invokes RTSP DESCRIBE request.

52. DUT sends 200 OK message and SDP information.

53. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

54. DUT sends 200 OK message and the media stream information.

55. ONVIF Client invokes RTSP PLAY request.

56. DUT sends 200 OK message and starts media streaming.

57. DUT sends JPEG RTP media stream to ONVIF Client over UDP. Verify that stream has JPEG encoding and [Width3, Height3] resolution.

58. DUT sends RTCP sender report to ONVIF Client.

59. DUT validates the received RTP and RTCP packets, decodes and renders them.

60. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

61. DUT sends 200 OK Response and terminates the RTSP Session.

62. Repeat steps 5-62 for the first, middle and last Video Encoder configurations supported by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

- DUT did not sent valid video stream with JPEG encoding and specified resolution or sent it in another encoding or resolution.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: The following rule will be used to compare resolutions: Resolution1 (Width1, Height1) is greater than Resolution2 (Width2, Height2), if Width1*Height1 is greater than Width2*Height2. This rule will be used to define minimum, maximum and middle resolution.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: If it is impossible to create profile for all three: the first, the middle and the last Video Encoder configurations on the 5th step, then Onvif Client will try selecting another Video Encoder configuration until a profile is created.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.24 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION

Test Case ID: RTSS-1-1-47

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri, Live Streaming

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of different Resolutions for video encoder configuration and video stream.

Pre-Requisite: MPEG4 is implemented by DUT. Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. There is Video Encoder Configuration which is either configured with MPEG4 or the DUT allows to change encoding of Video Encoder Configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.
4. Verify the **GetVideoEncoderConfigurationsResponse** message.
5. Find or create media profile with Video Source Configuration and Video Encoder Configuration with token VECToken1 and supporting of MPEG4 encoding, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message by following the procedure mentioned in Annex A.13. If it is not possible skip steps 6-61 and go to the step 62.
6. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “Profile1”, where “Profile1” is profile token from the step 5) to get video encoder configuration options.
7. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.
8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = MPEG4, where [Width1, Height1] is maximum resolution from the Options.MPEG4.ResolutionsAvailable) to change video encoder configuration.
9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.
10. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.
11. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = MPEG4) from the DUT. Check that new setting for Resolution and Encoding was applied.
12. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.
13. DUT sends RTSP URI in the **GetStreamUriResponse** message.
14. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
15. ONVIF Client invokes RTSP DESCRIBE request.
16. DUT sends 200 OK message and SDP information.

17. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

18. DUT sends 200 OK message and the media stream information.

19. ONVIF Client invokes RTSP PLAY request.

20. DUT sends 200 OK message and starts media streaming.

21. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP. Verify that stream has MPEG4 encoding and [Width1, Height1] resolution. (Note: if stream Height resolution is between Height1 – 8 and Height1 + 8 it is assumed as valid; if stream Width resolution is between Width1 – 8 and Width1 + 8 it is assumed as valid.)

22. DUT sends RTCP sender report to ONVIF Client.

23. DUT validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

25. DUT sends 200 OK Response and terminates the RTSP Session.

26. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width2, Height2], Encoding = MPEG4, where [Width2, Height2] is minimum resolution from the Options.MPEG4.ResolutionsAvailable) to change video encoder configuration.

27. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

28. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

29. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width2, Height2], Encoding = MPEG4, where [Width2, Height2]) from the DUT. Check that new setting for Resolution and Encoding was applied.

30. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

31. DUT sends RTSP URI in the **GetStreamUriResponse** message.

32. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

33. ONVIF Client invokes RTSP DESCRIBE request.

34. DUT sends 200 OK message and SDP information.

35. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

36. DUT sends 200 OK message and the media stream information.

37. ONVIF Client invokes RTSP PLAY request.

38. DUT sends 200 OK message and starts media streaming.

39. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP. Verify that stream has MPEG4 encoding and [Width2, Height2] resolution.

40. DUT sends RTCP sender report to ONVIF Client.

41. DUT validates the received RTP and RTCP packets, decodes and renders them.

42. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

43. DUT sends 200 OK Response and terminates the RTSP Session.

44. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width3, Height3], Encoding = MPEG4, where [Width3, Height3] is middle resolution from the Options.MPEG4.ResolutionsAvailable) to change video encoder configuration.

45. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

46. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

47. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width3, Height3], Encoding = MPEG4) from the DUT. Check that new setting for Resolution and Encoding was applied.

48. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

49. DUT sends RTSP URI in the **GetStreamUriResponse** message.

50. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

51. ONVIF Client invokes RTSP DESCRIBE request.

52. DUT sends 200 OK message and SDP information.

53. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

54. DUT sends 200 OK message and the media stream information.

55. ONVIF Client invokes RTSP PLAY request.

56. DUT sends 200 OK message and starts media streaming.

57. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP. Verify that stream has MPEG4 encoding and [Width3, Height3] resolution.

58. DUT sends RTCP sender report to ONVIF Client.

59. DUT validates the received RTP and RTCP packets, decodes and renders them.

60. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

61. DUT sends 200 OK Response and terminates the RTSP Session.

62. Repeat steps 5-62 for the first, middle and last Video Encoder configurations supported by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

- DUT did not sent valid video stream with MPEG4 encoding and specified resolution or sent it in another encoding or resolution.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: The following rule will be used to compare resolutions: Resolution1 (Width1, Height1) is greater than Resolution2 (Width2, Height2), if Width1*Height1 is greater than Width2*Height2. This rule will be used to define minimum, maximum and middle resolution.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: If it is impossible to create profile for all three: the first, the middle and the last Video Encoder configurations on the 5th step, then Onvif Client will try selecting another Video Encoder configuration until a profile is created.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.25 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION

Test Case ID: RTSS-1-1-48

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri, Live Streaming

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of different Resolutions for video encoder configuration and video stream.

Pre-Requisite: H.264 is implemented by DUT. Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. There is Video Encoder Configuration which is either configured with H.264 or the DUT allows to change encoding of Video Encoder Configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.

4. Verify the **GetVideoEncoderConfigurationsResponse** message.

5. Find or create media profile with Video Source Configuration and Video Encoder Configuration with token VECToken1 and supporting of H.264 encoding, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message by following the procedure mentioned in Annex A.13. If it is not possible skip steps 6-61 and go to the step 62.

6. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “Profile1”, where “Profile1” is profile token from the step 5) to get video encoder configuration options.

7. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.

8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = H.264, where [Width1, Height1] is maximum resolution from the Options.H.264.ResolutionsAvailable) to change video encoder configuration.

9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

10. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

11. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width1, Height1], Encoding = H.264) from the DUT. Check that new setting for Resolution and Encoding was applied.

12. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

13. DUT sends RTSP URI in the **GetStreamUriResponse** message.

14. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

15. ONVIF Client invokes RTSP DESCRIBE request.

16. DUT sends 200 OK message and SDP information.

17. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

18. DUT sends 200 OK message and the media stream information.

19. ONVIF Client invokes RTSP PLAY request.

20. DUT sends 200 OK message and starts media streaming.

21. DUT sends H.264 RTP media stream to ONVIF Client over UDP. Verify that stream has H.264 encoding and [Width1, Height1] resolution. (Note: if stream Height resolution is between Height1 – 8 and Height1 + 8 it is assumed as valid; if stream Width resolution is between Width1 – 8 and Width1 + 8 it is assumed as valid.)

22. DUT sends RTCP sender report to ONVIF Client.

23. DUT validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

25. DUT sends 200 OK Response and terminates the RTSP Session.

26. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width2, Height2], Encoding = H.264, where [Width2, Height2] is minimum resolution from the Options.H.264.ResolutionsAvailable) to change video encoder configuration.

27. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

28. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

29. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width2, Height2], Encoding = H.264, where [Width2, Height2]) from the DUT. Check that new setting for Resolution and Encoding was applied.

30. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

31. DUT sends RTSP URI in the **GetStreamUriResponse** message.

32. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

33. ONVIF Client invokes RTSP DESCRIBE request.

34. DUT sends 200 OK message and SDP information.

35. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

36. DUT sends 200 OK message and the media stream information.

37. ONVIF Client invokes RTSP PLAY request.

38. DUT sends 200 OK message and starts media streaming.

39. DUT sends H.264 RTP media stream to ONVIF Client over UDP. Verify that stream has H.264 encoding and [Width2, Height2] resolution.

40. DUT sends RTCP sender report to ONVIF Client.

41. DUT validates the received RTP and RTCP packets, decodes and renders them.

42. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

43. DUT sends 200 OK Response and terminates the RTSP Session.

44. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Resolution = [Width3, Height3], Encoding = H.264, where [Width3, Height3] is middle resolution from the Options.H.264.ResolutionsAvailable) to change video encoder configuration.

45. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

46. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

47. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = VECToken1, Resolution = [Width3, Height3], Encoding = H.264) from the DUT. Check that new setting for Resolution and Encoding was applied.

48. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

49. DUT sends RTSP URI in the **GetStreamUriResponse** message.

50. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

51. ONVIF Client invokes RTSP DESCRIBE request.

52. DUT sends 200 OK message and SDP information.

53. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

54. DUT sends 200 OK message and the media stream information.

55. ONVIF Client invokes RTSP PLAY request.

56. DUT sends 200 OK message and starts media streaming.

57. DUT sends H.264 RTP media stream to ONVIF Client over UDP. Verify that stream has H.264 encoding and [Width3, Height3] resolution.

58. DUT sends RTCP sender report to ONVIF Client.

59. DUT validates the received RTP and RTCP packets, decodes and renders them.

60. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

61. DUT sends 200 OK Response and terminates the RTSP Session.

62. Repeat steps 5-62 for the first, middle and last Video Encoder configurations supported by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

- DUT did not sent valid video stream with H.264 encoding and specified resolution or sent it in another encoding or resolution.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: The following rule will be used to compare resolutions: Resolution1 (Width1, Height1) is greater than Resolution2 (Width2, Height2), if Width1*Height1 is greater than Width2*Height2. This rule will be used to define minimum, maximum and middle resolution.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: If it is impossible to create profile for all three: the first, the middle and the last Video Encoder configurations on the 5th step, then Onvif Client will try selecting another Video Encoder configuration until a profile is created.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.26 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS)

Test Case ID: RTSS-1-1-49

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP-Unicast/UDP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

9. DUT sends RTSP URI in the **GetStreamUriResponse** message.

10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

11. ONVIF Client invokes RTSP DESCRIBE request.

12. DUT sends 200 OK message and SDP information.

13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

14. DUT sends 200 OK message and the media stream information.

15. ONVIF Client invokes RTSP PLAY request.

16. DUT sends 200 OK message and starts media streaming.

17. DUT sends JPEG RTP media stream to ONVIF Client over UDP.

18. DUT sends RTCP sender report to ONVIF Client.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. Repeat steps 8-20 to start video streaming for all profiles from step 7.

21. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

22. DUT sends 200 OK Response and terminates the RTSP Session.

23. Repeat steps 5-23 for the rest Video Source Configuration.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.27 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Unicast/RTSP/HTTP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS)

Test Case ID: RTSS-1-1-50

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP-Unicast/RTSP/HTTP/TCP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the first media profile from step 7.

9. DUT sends HTTP URI in the **GetStreamUriResponse** message.

10. ONVIF Client verifies the HTTP media stream URI provided by the DUT.

11. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

12. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

13. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

14. DUT sends 200 OK message and SDP information on HTTP GET connection.

15. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter for both audio and video streams separately.

16. DUT sends 200 OK message and the media stream information on HTTP GET connection.

17. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

18. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

19. DUT transfers RTP media stream to ONVIF Client on HTTP GET connection.

20. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

21. DUT validates the received RTP and RTCP packets, decodes and renders them.

22. Repeat steps 8-22 to start video streaming for all profiles from step 7.

23. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

24. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

25. Repeat steps 5-25 for the rest video source configuration.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

- DUT did not send valid video stream.

- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.28 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP) (ALL VIDEO SOURCE CONFIGURATIONS)

Test Case ID: RTSS-1-1-51

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP/RTSP/TCP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. RTP/RTSP/TCP media streaming is implemented by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.
4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.
5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source

configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.
7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.
8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the first media profile from step 7.
9. DUT sends RTSP URI in the **GetStreamUriResponse** message.
10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
11. ONVIF Client invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/TCP.
14. DUT sends 200 OK message and the media stream information.
15. ONVIF Client invokes RTSP PLAY request.
16. DUT sends 200 OK message and starts media streaming.
17. DUT sends JPEG RTP media stream to ONVIF Client over TCP.
18. DUT sends RTCP sender report to ONVIF Client.
19. DUT validates the received RTP and RTCP packets, decodes and renders them.
20. Repeat steps 8-20 to start video streaming for all profiles from step 7.
21. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.
22. DUT sends 200 OK Response and terminates the RTSP Session.
23. Repeat steps 5-23 for the rest Video Source Configuration.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.29 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES) (ALL VIDEO SOURCE CONFIGURATIONS)

Test Case ID: RTSS-1-1-52

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (Mix Of Transport Types)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. RTP/RTSP/TCP media streaming is implemented by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.
4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.
5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.
6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.
7. For each Media Profile *profile* that will be used for multicast streaming:

   7.1. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile
   - in *profile* - Media Profile to update
8. ONVIF Client start media streaming for first profile to retrieve media stream URI for the first media profile from step 7. (To start media streaming Annex A.15, Annex A.16, Annex A.17, or Annex A.18 will be used in loop. Annex A.16 will be used if RTP-Multicast/UDP supported. Annex A.18 will be used if RTP/RTSP/TCP supported.)
9. Repeat step 8 to start video streaming for all profiles from step 7.
10. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

11. DUT sends 200 OK Response and terminates the RTSP Session.

12. Repeat steps 5-12 for the rest video source configuration.

13. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: In case if there are more than one Media Profiles with the same Video Encoder Configuration, multicast streaming will be started only for one of them.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

5.1.1.30 MEDIA STREAMING – JPEG (VALIDATING RTP HEADER EXTENSION)

Test Case ID: RTSS-1-1-53

Specification Coverage: JPEG over RTP

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To validate JPEG RTP header extension for devices supporting high resolutions for JPEG.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. A media profile with JPEG video encoder configuration exists The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4 and Resolution = [“Width”, “Height”] shall be enough to use JPEG header extension (greater than 2 MP). If there is no such resolutions use any Resolution.

5. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

6. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

7. DUT sends RTSP URI in the **GetStreamUriResponse** message.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request.

10. DUT sends 200 OK message and SDP information.

11. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

12. DUT sends 200 OK message and the media stream information.

13. ONVIF Client invokes RTSP PLAY request.

14. DUT sends 200 OK message and starts media streaming.

15. DUT sends JPEG RTP media stream to ONVIF Client over UDP.

16. DUT sends RTCP sender report to ONVIF Client.

17. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. DUT validates RTP header extensions for each packet. If resolution was greater than 2MP check that JPEG header extension is present and valid. If resolution was less than 2MP check that JPEG header extension is valid if present.

19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

20. DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid RTP header extension in one or more media streams.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.31 MEDIA STREAMING – JPEG (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-1-1-54

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify JPEG media streaming based on RTP/UDP Unicast Transport for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. A media profile with JPEG video encoder configuration. IPv6 is turned ON for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.
8. DUT sends RTSP URI in the **GetStreamUriResponse** message.
9. Verify the RTSP media stream URI provided by the DUT.
10. ONVIF Client invokes RTSP DESCRIBE request.
11. DUT sends 200 OK message and SDP information.
12. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.
13. DUT sends 200 OK message and the media stream information.
14. ONVIF Client invokes RTSP PLAY request.
15. DUT sends 200 OK message and starts media streaming.
16. DUT sends JPEG RTP media stream to ONVIF Client over UDP.
17. DUT sends RTCP sender report to ONVIF Client.
18. DUT validates the received RTP and RTCP packets, decodes and renders them.
19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
20. DUT sends 200 OK Response and terminates the RTSP Session.
21. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.

- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.32 MEDIA STREAMING – JPEG (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-1-1-55

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify JPEG media streaming based on HTTP Transport for IPv6

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. A media profile with JPEG video encoder configuration. IPv6 is turned ON for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.

8. DUT sends HTTP URI in the **GetStreamUriResponse** message.

9. ONVIF Client verifies the HTTP media stream URI provided by the DUT.

10. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

11. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

12. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

13. DUT sends 200 OK message and SDP information on HTTP GET connection.

14. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

15. DUT sends 200 OK message and the media stream information on HTTP GET connection.

16. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

17. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

18. DUT transfers JPEG RTP media stream to ONVIF Client on HTTP GET connection.

19. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

20. DUT validates the received RTP and RTCP packets, decodes and renders them.

21. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

22. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

23. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.33 MEDIA STREAMING – JPEG (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-1-1-56

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify JPEG media streaming based on RTP/RTSP/TCP using RTSP tunnel for IPv6.

Pre-Requisite: : Media Service was received from the DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP media streaming is implemented by DUT. A media profile with JPEG video encoder configuration. IPv6 is turned ON for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.
6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.
8. DUT sends RTSP URI in the **GetStreamUriResponse** message.
9. Verify the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.

17. 16. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

19. DUT sends 200 OK Response and terminates the RTSP Session.

20. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.34 MEDIA STREAMING – MPEG4 (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-1-1-57

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on RTP/UDP Unicast Transport for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. MPEG4 is implemented by DUT. A media profile with MPEG4 video encoder configuration. IPv6 is turned ON for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1

and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

8. DUT sends RTSP URI in the **GetStreamUriResponse** message.

9. Verify the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. DUT sends MPEG4 RTP media stream to ONVIF Client over UDP.

17. DUT sends RTCP sender report to ONVIF Client.

18. DUT validates the received RTP and RTCP packets, decodes and renders them.

19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

20. DUT sends 200 OK Response and terminates the RTSP Session.

21. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.35 MEDIA STREAMING – MPEG4 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-1-1-58

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on HTTP Transport for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. MPEG4 is implemented by DUT. A media profile with MPEG4 video encoder configuration. IPv6 is turned ON for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.

8. DUT sends HTTP URI in the **GetStreamUriResponse** message.

9. ONVIF Client verifies the HTTP media stream URI provided by the DUT.

10. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

11. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

12. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.

13. DUT sends 200 OK message and SDP information on HTTP GET connection.

14. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

15. DUT sends 200 OK message and the media stream information on HTTP GET connection.

16. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

17. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

18. DUT transfers MPEG4 RTP media stream to ONVIF Client on HTTP GET connection.

19. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

20. DUT validates the received RTP and RTCP packets, decodes and renders them.

21. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

22. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

23. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.36 MEDIA STREAMING – MPEG4 (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-1-1-59

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on RTP/RTSP/TCP using RTSP tunnel for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. MPEG4 and RTP/RTSP/TCP media streaming is implemented by DUT. A media profile with MPEG4 video encoder configuration. IPv6 is turned ON for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.
6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.

8. DUT sends RTSP URI in the **GetStreamUriResponse** message.

9. Verify the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.

17. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

19. DUT sends 200 OK Response and terminates the RTSP Session.

20. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.37 MEDIA STREAMING – H.264 (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-1-1-60

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on RTP/UDP Unicast Transport for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. H.264 is implemented by DUT. A media profile with H.264 video encoder configuration. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, UDP transport) to retrieve media stream URI for the selected media profile.

8. DUT sends RTSP URI in the **GetStreamUriResponse** message.

9. Verify the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP/UDP.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. DUT sends H.264 RTP media stream to ONVIF Client over UDP.

17. DUT sends RTCP sender report to ONVIF Client.

18. DUT validates the received RTP and RTCP packets, decodes and renders them.

19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

20. DUT sends 200 OK Response and terminates the RTSP Session.

21. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.38 MEDIA STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-1-1-61

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on HTTP Transport for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. H.264 is implemented by DUT. A media profile with H.264 video encoder configuration. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.
6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.
7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, HTTP transport) to retrieve media stream URI for the selected media profile.
8. DUT sends HTTP URI in the **GetStreamUriResponse** message.
9. ONVIF Client verifies the HTTP media stream URI provided by the DUT.
10. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.
11. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.
12. ONVIF Client invokes RTSP DESCRIBE request on HTTP POST connection.
13. DUT sends 200 OK message and SDP information on HTTP GET connection.
14. ONVIF Client invokes RTSP SETUP request on HTTP POST connection with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.
15. DUT sends 200 OK message and the media stream information on HTTP GET connection.

16. ONVIF Client invokes RTSP PLAY request on HTTP POST connection.

17. DUT sends 200 OK message and starts media streaming on HTTP GET connection.

18. DUT transfers H.264 RTP media stream to ONVIF Client on HTTP GET connection.

19. DUT sends RTCP sender report to ONVIF Client on HTTP GET connection.

20. DUT validates the received RTP and RTCP packets, decodes and renders them.

21. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection and closes the HTTP POST connection.

22. DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

23. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.1.39 MEDIA STREAMING – H.264 (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-1-1-62

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

Command Under Test: None.

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on RTP/RTSP/TCP using RTSP tunnel for IPv6.

Pre-Requisite: Media Service was received from the DUT. Real-time streaming supported by DUT. H.264 and RTP/RTSP/TCP media streaming is implemented by DUT. A media profile with H.264 video encoder configuration. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile with H.264 video encoding support by following the procedure mentioned in Annex A.4.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Session Timeout = t1 and force persistence = false). These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse** in Annex A.4.

6. DUT modifies video encoder configuration and responds with **SetVideoEncoderConfigurationResponse** message indicating success.

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Unicast, RTSP transport) to retrieve media stream URI for the selected media profile.

8. DUT sends RTSP URI in the **GetStreamUriResponse** message.

9. Verify the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as ‘RTP/TCP’ along with ‘interleaved’ parameter.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. DUT interleaves RTP and RTCP packets, send them over RTSP control connection.

17. DUT validates the received RTP and RTCP packets, decodes and renders them.

18. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

19. DUT sends 200 OK Response and terminates the RTSP Session.

20. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not have valid media profile.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send RTP and RTCP packets as per [RFC 2326] section 10.12.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2 Multicast

5.1.2.1 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP)

Test Case ID: RTSS-1-2-12

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances.

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP-Multicast/UDP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. For each Media Profile (*profile*) that will be used for streaming:

   8.1. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

   - in *profile* - Media Profile to update

9. Configure multicast settings for Multicast Address from profile (see Annex A.11).

10. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

11. DUT sends RTSP URI in the **GetStreamUriResponse** message.

12. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

13. ONVIF Client invokes RTSP DESCRIBE request.

14. DUT sends 200 OK message and SDP information.

15. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/UDP.

16. DUT sends 200 OK message and the media stream information.

17. ONVIF Client invokes RTSP PLAY request.

18. DUT sends 200 OK message and starts media streaming.

19. DUT sends JPEG RTP multicast media stream to ONVIF Client over UDP.

20. DUT sends RTCP sender report to ONVIF Client.

21. DUT validates the received RTP and RTCP packets, decodes and renders them.

22. Repeat steps 8-20 to start video streaming for all profiles from step 7.

23. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

24. DUT sends 200 OK Response and terminates the RTSP Session.

25. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than one, repeat steps 5-23 for the last video source configuration token from **GetVideoEncoderConfigurationsResponse** message.

26. If number of Video source configurations in **GetVideoSourceConfigurationsResponse** message is more than two, repeat steps 5-23 for the video source configuration token between the first and the last VSC tokens from **GetVideoSourceConfigurationsResponse** message.

27. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: In case if there are more than one Media Profiles with the same Video Encoder Configuration, multicast streaming will be started only for one of them.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.2 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-1-2-13

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, JPEG over RTP, Stream control, RTSP.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify JPEG media streaming based on RTP-Multicast/UDP Transport for IPv4.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. A media profile with JPEG video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile (*profile*) with JPEG video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings.

5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

6. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

   - in *profile* - Media Profile to update

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

8. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

9. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/UDP.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. The DUT sends JPEG RTP multicast media stream to multicast IPv4 address over UDP.

17. DUT sends RTCP sender report to ONVIF Client.

18. DUT validates the received RTP and RTCP packets, decodes and renders them.

19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

20. DUT sends 200 OK Response and terminates the RTSP Session.

21. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- DUT did not send valid video stream.

- The DUT did not send JPEG RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.3 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-1-2-14

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, MPEG4 over RTP, Stream control, RTSP.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on RTP-Multicast/UDP Transport for IPv4.

Pre-Requisite: Media is supported by DUT and MPEG4 is implemented by DUT. Real-time streaming supported by DUT. A media profile with MPEG4 video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile (*profile*) with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.
4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set MPEG4 encoding and Multicast settings.
5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.
6. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters
   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile
   - in *profile* - Media Profile to update

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

8. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

9. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/ UDP.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. The DUT sends MPEG4 RTP multicast media stream to multicast IPv4 address over UDP.

17. DUT sends RTCP sender report to ONVIF Client.

18. DUT validates the received RTP and RTCP packets, decodes and renders them.

19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

20. DUT sends 200 OK Response and terminates the RTSP Session.

21. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- DUT did not send valid video stream.
- The DUT did not send MPEG4 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.4 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-1-2-15

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, H.264 over RTP, Stream control, RTSP.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on RTP-Multicast/UDP Transport for IPv4.

Pre-Requisite: Media is supported by DUT and H.264 is implemented by DUT. Real-time streaming supported by DUT. A media profile with H.264 video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile (*profile*) with H.264 video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set H.264 encoding and Multicast settings.

5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

6. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

   - in *profile* - Media Profile to update

7. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

8. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

9. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

10. ONVIF Client invokes RTSP DESCRIBE request.

11. DUT sends 200 OK message and SDP information.

12. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/UDP.

13. DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request.

15. DUT sends 200 OK message and starts media streaming.

16. The DUT sends H.264 RTP multicast media stream to multicast IPv4 address over UDP.

17. DUT sends RTCP sender report to ONVIF Client.

18. DUT validates the received RTP and RTCP packets, decodes and renders them.

19. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

20. DUT sends 200 OK Response and terminates the RTSP Session.

21. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- DUT did not send valid video stream.

- The DUT did not send H.264 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.5 MEDIA STREAMING – JPEG (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-1-2-16

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, JPEG over RTP, Stream control, RTSP.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify JPEG media streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. A media profile with JPEG video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. IPv6 network interface is turned on. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client selects a media profile (*profile*) with JPEG video encoding support by following the procedure mentioned in Annex A.4.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings.

6. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

7. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

- in *profile* - Media Profile to update

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

9. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

11. ONVIF Client invokes RTSP DESCRIBE request.

12. DUT sends 200 OK message and SDP information.

13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/UDP.

14. DUT sends 200 OK message and the media stream information.

15. ONVIF Client invokes RTSP PLAY request.

16. DUT sends 200 OK message and starts media streaming.

17. The DUT sends JPEG RTP multicast media stream to multicast IPv4 address over UDP.

18. DUT sends RTCP sender report to ONVIF Client.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

21. DUT sends 200 OK Response and terminates the RTSP Session.

22. The DTT restores changed profiles and configurations.

23. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- DUT did not send valid video stream.
- The DUT did not send JPEG RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.6 MEDIA STREAMING – MPEG4 (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-1-2-17

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, MPEG4 over RTP, Stream control, RTSP.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify MPEG4 media streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT and MPEG4 is implemented by DUT. Real-time streaming supported by DUT. A media profile with MPEG4 video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile (*profile*) with MPEG4 video encoding support by following the procedure mentioned in Annex A.4.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “MPEG4”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set MPEG4 encoding and Multicast settings.
6. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.
7. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters
   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile
   - in *profile* - Media Profile to update
8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.
9. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
11. ONVIF Client invokes RTSP DESCRIBE request.
12. DUT sends 200 OK message and SDP information.
13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/ UDP.

14. DUT sends 200 OK message and the media stream information.

15. ONVIF Client invokes RTSP PLAY request.

16. DUT sends 200 OK message and starts media streaming.

17. The DUT sends MPEG4 RTP multicast media stream to multicast IPv4 address over UDP.

18. DUT sends RTCP sender report to ONVIF Client.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

21. DUT sends 200 OK Response and terminates the RTSP Session.

22. The DTT restores changed profiles and configurations.

23. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- DUT did not send **GetStreamUriResponse** message.

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

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

- DUT did not send RTCP sender report correctly.

- DUT did not send valid video stream.

- The DUT did not send MPEG4 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.7 MEDIA STREAMING – H.264 (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-1-2-18

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, H.264 over RTP, Stream control, RTSP.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify H.264 media streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT and H.264 is implemented by DUT. Real-time streaming supported by DUT. A media profile with H.264 video encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile (*profile*) with H.264 video encoding support by following the procedure mentioned in Annex A.4.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “H.264”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”,

"multicastAddress1"], Multicast.Port = "port1", TTL = "ttl1", Session Timeout = t1 and force persistence = false) to set H.264 encoding and Multicast settings.

6. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

7. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

   - in *profile* - Media Profile to update

8. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

9. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

11. ONVIF Client invokes RTSP DESCRIBE request.

12. DUT sends 200 OK message and SDP information.

13. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/UDP.

14. DUT sends 200 OK message and the media stream information.

15. ONVIF Client invokes RTSP PLAY request.

16. DUT sends 200 OK message and starts media streaming.

17. The DUT sends H.264 RTP multicast media stream to multicast IPv4 address over UDP.

18. DUT sends RTCP sender report to ONVIF Client.

19. DUT validates the received RTP and RTCP packets, decodes and renders them.

20. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.

21. DUT sends 200 OK Response and terminates the RTSP Session.

22. The DTT restores changed profiles and configurations.

23. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- DUT did not send valid video stream.
- The DUT did not send H.264 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.8 VIDEO ENCODER CONFIGURATION – MULTICAST PORT (IPv4)

Test Case ID: RTSS-1-2-19

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, Start multicast streaming, Stop multicast streaming

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of Multicast port and address for Video Encoder Configuration in case of the same Multicast addresses and different Multicast ports for all Video Encoder Configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.

4. Verify the **GetVideoEncoderConfigurationsResponse** message.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message) to change multicast port and address for video encoder configuration.

6. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

7. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

8. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = Token1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

9. Repeat steps 5-8 for the rest Video Encoder configurations supported by the DUT with using different multicast ports and the same multicast addresses for Video Encoder Configurations.

10. Find or create media profile (*profile*) with Video Source Configuration and Video Encoder Configuration with token VECToken1 (see Annex A.12). If it is not possible skip steps 11-19 and go to the step 20.

11. ONVIF Client removes Audio Encoder Configuration, Audio Source Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

    - in { *AudioEncoderConfiguration, AudioSourceConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

    - in *profile* - Media Profile to update

12. Configure multicast settings for other entities from profile if required (see Annex A.11).

13. ONVIF Client invokes **StartMulticastStreamingRequest** message (ProfileToken = [profile token from the step 10]) to start multicast streaming.

14. Verify the **StartMulticastStreamingResponse** from the DUT.

15. The DUT sends video RTP multicast media stream to multicast IPv4 address over UDP.

16. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

17. ONVIF Client validates that specified multicast address and port are used.

18. ONVIF Client invokes **StopMulticastStreamingRequest** message (ConfigurationToken ProfileToken = [profile token from the step 10]) to stop multicast streaming from specified port.

19. Verify the **StopMulticastStreamingResponse** from the DUT.

20. Repeat steps 10-19 for the rest Video Encoder configuration supported by the DUT.

21. Restore Video Encoder Configurations settings and media profiles.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetVideoEncoderConfigurationsResponse** message.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send valid **GetVideoEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during steps 5-6.
- The DUT did not send valid **StartMulticastStreamingResponse** message.
- The DUT did not send valid video stream to specified multicast address and port.
- The DUT did not send valid **StopMulticastStreamingResponse** message.
- It is not possible to find or create profile for all Video Encoder Configurations.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.9 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4)

Test Case ID: RTSS-1-2-20

Specification Coverage: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, Start multicast streaming, Stop multicast streaming

Command Under Test: SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of Multicast port and address for Video Encoder Configuration in case of the same Multicast port and different Multicast address for all Video Encoder Configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetVideoEncoderConfigurationsRequest** message to retrieve video configuration list.

4. Verify the **GetVideoEncoderConfigurationsResponse** message.

5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, where VECToken1 is first video encoder configuration token from **GetVideoEncoderConfigurationsResponse** message) to change multicast port and address for video encoder configuration.

6. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

7. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message (ConfigurationToken = VECToken1) to get video encoder configuration.

8. Verify the **GetVideoEncoderConfigurationResponse** message (ConfigurationToken = Token1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

9. Repeat steps 5-8 for the rest Video Encoder configurations supported by the DUT with using different multicast addresses and the same multicast port for Video Encoder Configurations.

10. Find or create media profile (*profile*) with Video Source Configuration and Video Encoder Configuration with token VECToken1 (see Annex A.12). If it is not possible skip steps 11-19 and go to the step 20.

11. ONVIF Client removes Audio Encoder Configuration, Audio Source Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

    - in { *AudioEncoderConfiguration, AudioSourceConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

    - in *profile* - Media Profile to update

12. Configure multicast settings for other entities from profile if required (see Annex A.11).

13. ONVIF Client invokes **StartMulticastStreamingRequest** message (ProfileToken = [profile token from the step 10]) to start multicast streaming.

14. Verify the **StartMulticastStreamingResponse** from the DUT.

15. The DUT sends video RTP multicast media stream to multicast IPv4 address over UDP.

16. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

17. ONVIF Client validates that specified multicast address and port are used.

18. ONVIF Client invokes **StopMulticastStreamingRequest** message (ConfigurationToken ProfileToken = [profile token from the step 10]) to stop multicast streaming from specified port.

19. Verify the **StopMulticastStreamingResponse** from the DUT.

20. Repeat steps 10-19 for the rest Video Encoder configuration supported by the DUT.

21. Restore Video Encoder Configurations settings and media profiles.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetVideoEncoderConfigurationsResponse** message.

- DUT did not send **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send valid **GetVideoEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during steps 5-6.

- The DUT did not send valid **StartMulticastStreamingResponse** message.

- The DUT did not send valid video stream to specified multicast address and port.

- The DUT did not send valid **StopMulticastStreamingResponse** message.

- It is not possible to find or create profile for all Video Encoder Configurations.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.10 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv4)

Test Case ID: RTSS-1-2-21

Specification Coverage: Multicast streaming

Command Under Test: none

WSDL Reference: media.wsdl

Test Purpose: To verify possibility to use multicast address and port during RTSP SETUP as set in Video Encoder Configuration (IPv4).

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. RTP-Multicast/UDP transport protocol is supported by the DUT. A media profile with JPEG video encoder configuration exists. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile (*profile*) with JPEG video encoding support by following the procedure mentioned in Annex A.19.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings.

5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

6. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

- in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile
- in *profile* - Media Profile to update

7. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.
8. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
9. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
10. ONVIF Client invokes RTSP DESCRIBE request.
11. The DUT sends 200 OK message and SDP information.
12. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP and destination = multicastAddress1, port = port1-port2.
13. The DUT sends 200 OK message and the media stream information. Verify that destination = multicastAddress1, port = port1-port2 was received.
14. ONVIF Client invokes RTSP PLAY request.
15. The DUT sends 200 OK message and starts media streaming.
16. The DUT sends JPEG RTP multicast media stream to multicast IPv4 address over UDP.
17. Verify that specified multicast port and address are used for streaming.
18. The DUT sends RTCP sender report to ONVIF Client.
19. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.
20. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
21. The DUT sends 200 OK Response and terminates the RTSP Session.
22. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetVideoEncoderConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid Multicast address in RTSP 200 OK response for RTSP SETUP requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send JPEG RTP multicast media streaming for corresponding multicast IP and port.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.11 VIDEO ENCODER CONFIGURATION – MULTICAST ADDRESS AND PORT IN RTSP SETUP (IPv6)

Test Case ID: RTSS-1-2-22

Specification Coverage: Multicast streaming

Command Under Test: none

WSDL Reference: media.wsdl

Test Purpose: To verify possibility to use multicast address and port during RTSP SETUP as set in Video Encoder Configuration (IPv6).

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. RTP-Multicast/UDP transport protocol is supported by the DUT. A media profile with JPEG video encoder configuration exists. IPv6 is supported by the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client selects a media profile (*profile*) with JPEG video encoding support by following the procedure mentioned in Annex A.19.
5. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv6”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings.
6. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.
7. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters
   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile
   - in *profile* - Media Profile to update
8. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.
9. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
10. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

11. ONVIF Client invokes RTSP DESCRIBE request.

12. The DUT sends 200 OK message and SDP information.

13. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP and destination = multicastAddress1, port = port1-port2.

14. The DUT sends 200 OK message and the media stream information. Verify that destination = multicastAddress1, port = port1-port2 was received.

15. ONVIF Client invokes RTSP PLAY request.

16. The DUT sends 200 OK message and starts media streaming.

17. The DUT sends JPEG RTP multicast media stream to multicast IPv6 address over UDP.

18. Verify that specified multicast port and address are used for streaming.

19. The DUT sends RTCP sender report to ONVIF Client.

20. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

21. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

22. The DUT sends 200 OK Response and terminates the RTSP Session.

23. The DTT restores changed profiles and configurations.

24. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

- The DUT did not send valid Multicast address in RTSP 200 OK response for RTSP SETUP requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send JPEG RTP multicast media streaming for corresponding multicast IP and port.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.1.2.12 MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP-Multicast/UDP) (ALL VIDEO SOURCE CONFIGURATIONS)

Test Case ID: RTSS-1-2-23

Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances.

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Guaranteed Number Of Video Encoder Instances (RTP-Multicast/UDP)

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client will invoke **GetVideoSourceConfigurationsRequest** message to retrieve all DUT video source configurations.

4. Verify the **GetVideoSourceConfigurationsResponse** message from the DUT.

5. ONVIF Client will invoke **GetGuaranteedNumberOfVideoEncoderInstancesRequest** message (ConfigurationToken = “VSCToken1”, where “VSCToken1” is a first video source configuration token from **GetVideoSourceConfigurationsResponse** message) to retrieve guaranteed number of video encoder instances per first video source configuration.

6. Verify the **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message from the DUT.

7. Create or find number of profiles equal to TotalNumber from **GetGuaranteedNumberOfVideoEncoderInstancesResponse** message that contains video source configuration with token “VSCToken1” and video encoder configuration by following the procedure mentioned in Annex A.14.

8. For each Media Profile (*profile*) that will be used for streaming:

   8.1. ONVIF Client removes Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

   - in { *AudioEncoderConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

   - in *profile* - Media Profile to update

9. Configure multicast settings for Multicast Address from profile (see Annex A.11).

10. ONVIF Client invokes **GetStreamUriRequest** message (Profile Token, RTP-Multicast, UDP transport) to retrieve media stream URI for the first media profile from step 7.

11. DUT sends RTSP URI in the **GetStreamUriResponse** message.

12. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

13. ONVIF Client invokes RTSP DESCRIBE request.

14. DUT sends 200 OK message and SDP information.

15. ONVIF Client invokes RTSP SETUP request with transport parameter as RTP-Multicast/UDP.
16. DUT sends 200 OK message and the media stream information.
17. ONVIF Client invokes RTSP PLAY request.
18. DUT sends 200 OK message and starts media streaming.
19. DUT sends JPEG RTP multicast media stream to ONVIF Client over UDP.
20. DUT sends RTCP sender report to ONVIF Client.
21. DUT validates the received RTP and RTCP packets, decodes and renders them.
22. Repeat steps 8-20 to start video streaming for all profiles from step 7.
23. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session for each started stream.
24. DUT sends 200 OK Response and terminates the RTSP Session.
25. Repeat steps 5-23 for the rest video source configuration.
26. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **GetStreamUriResponse** message.
- DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- DUT did not send valid RTP header in one or more media streams.
- DUT did not send RTCP sender report correctly.
- RTSP Session is terminated by DUT during media streaming.
- DUT did not send valid video stream.
- DUT does not provide possibility to create TotalNumber of profiles for step 7.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: In case if there are more than one Media Profiles with the same Video Encoder Configuration, multicast streaming will be started only for one of them.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2 Audio Streaming

5.2.1 Unicast

5.2.1.1 MEDIA STREAMING – G.711 (RTP-Unicast/UDP)

Test Case ID: RTSS-2-1-19

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/UDP transport when media profile contains video and audio configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

    - in G711 - required audio encoding

    - in UDP - Transport Protocol

    - in RTP-Unicast - Stream Type

    - in IPv4 - IP version

    - out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

    - in *streamUri* - Uri for media streaming

    - in audio - media type

    - in G.711 - expected media stream encoding

5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.2 MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-2-1-20

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters
   - in G711 - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming
4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in G.711 - expected media stream encoding
5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.3 MEDIA STREAMING – G.711 (RTP/RTSP/TCP)

Test Case ID: RTSS-2-1-21

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP/RTSP/TCP Transport.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Audio is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in G711 - required audio encoding

   - in HTTP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv4 - IP version

   - out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

- in G.711 - expected media stream encoding

5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.4 MEDIA STREAMING – G.726 (RTP-Unicast/UDP)

Test Case ID: RTSS-2-1-22

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/UDP transport when media profile contains video and audio configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in G726 - required audio encoding

   - in UDP - Transport Protocol

- in RTP-Unicast - Stream Type
- in IPv4 - IP version
- out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in G.726 - expected media stream encoding
5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.5 MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-2-1-23

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in G726 - required audio encoding

   - in HTTP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv4 - IP version

   - out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

   - in G.726 - expected media stream encoding

5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.6 MEDIA STREAMING – G.726 (RTP/RTSP/TCP)

Test Case ID: RTSS-2-1-24

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP/RTSP/TCP Transport.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in G726 - required audio encoding

   - in HTTP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv4 - IP version

   - out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

   - in G.726 - expected media stream encoding

5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.7 MEDIA STREAMING – AAC (RTP-Unicast/UDP)

Test Case ID: RTSS-2-1-25

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/UDP transport when media profile contains video and audio configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in AAC - required audio encoding

   - in UDP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv4 - IP version

   - out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

- in *streamUri* - Uri for media streaming
- in audio - media type
- in AAC - expected media stream encoding

5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.8 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-2-1-26

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

- in AAC - required audio encoding
- in HTTP - Transport Protocol
- in RTP-Unicast - Stream Type
- in IPv4 - IP version
- out *streamUri* - Uri for media streaming

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in AAC - expected media stream encoding
5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.9 MEDIA STREAMING – AAC (RTP/RTSP/TCP)

Test Case ID: RTSS-2-1-27

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP/RTSP/TCP Transport.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters
   - in AAC - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming
4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in AAC - expected media stream encoding
5. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.10 AUDIO STREAMING – G.711 (RTP-Unicast/UDP)

Test Case ID: RTSS-2-1-28

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/UDP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (Action/MaxNVTProfiles or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.
9. DUT sends the list of audio encoder configurations compatible with the received media profile token.
10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.
11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as 'testprofileX' and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = "G.711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = "RTP-Unicast", Transport.Protocol = "UDP") to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/UDP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends G.711 RTP media stream to ONVIF Client over UDP

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.711 RTP media streaming to ONVIF Client.
- The DUT sent not only G.711 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.11 AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-2-1-29

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (Action/MaxNVTProfiles or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioSourceConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.711”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “HTTP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/RTSP/HTTP/TCP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends G.711 RTP media stream to ONVIF Client over HTTP.

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send G.711 RTP media streaming to ONVIF Client.

- The DUT sent not only G.711 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.12 AUDIO STREAMING – G.711 (RTP/RTSP/TCP)

Test Case ID: RTSS-2-1-30

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP/RTSP/TCP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.711”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP/RTSP/TCP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends G.711 RTP media stream to ONVIF Client over RTSP

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send G.711 RTP media streaming to ONVIF Client.
- The DUT sent not only G.711 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.13 AUDIO STREAMING – G.726 (RTP-Unicast/UDP)

Test Case ID: RTSS-2-1-31

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/UDP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify CreateProfileResponse message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from GetCompatibleAudioEncoderConfigurationsResponse message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.726”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/UDP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends G.726 RTP media stream to ONVIF Client over UDP

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP media streaming to ONVIF Client.
- The DUT sent not only G.726 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.14 AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-2-1-32

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioSourceConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.726”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “HTTP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/RTSP/HTTP/TCP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends G.726 RTP media stream to ONVIF Client over HTTP.

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP media streaming to ONVIF Client.
- The DUT sent not only G.726 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.15 AUDIO STREAMING – G.726 (RTP/RTSP/TCP)

Test Case ID: RTSS-2-1-33

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP/RTSP/TCP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.
9. DUT sends the list of audio encoder configurations compatible with the received media profile token.
10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.
11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.
12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.726”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP/RTSP/TCP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends G.726 RTP media stream to ONVIF Client over RTSP

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP media streaming to ONVIF Client.
- The DUT sent not only G.726 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.16 AUDIO STREAMING – AAC (RTP-Unicast/UDP)

Test Case ID: RTSS-2-1-34

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/UDP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = 'testprofileX'.

4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = 'testprofileX') to retrieve the list of audio source configurations compatible with profile.

6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = 'testprofileX' and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = 'testprofileX') to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/UDP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends AAC RTP media stream to ONVIF Client over UDP

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send AAC RTP media streaming to ONVIF Client.
- The DUT sent not only AAC RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.17 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-2-1-35

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (Action/MaxNVTProfiles or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioSourceConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “HTTP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/RTSP/HTTP/TCP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends AAC RTP media stream to ONVIF Client over HTTP.

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send AAC RTP media streaming to ONVIF Client.
- The DUT sent not only AAC RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.18 AUDIO STREAMING – AAC (RTP/RTSP/TCP)

Test Case ID: RTSS-2-1-36

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP/RTSP/TCP Transport for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 6-12 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

15. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.

16. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

17. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

18. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

19. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

20. ONVIF Client invokes RTSP DESCRIBE request.

21. The DUT sends 200 OK message and SDP information.

22. ONVIF Client invokes RTSP SETUP request with transport parameter RTP/RTSP/TCP.

23. The DUT sends 200 OK message and the media stream information.

24. ONVIF Client invokes RTSP PLAY request.

25. The DUT sends 200 OK message and starts media streaming.

26. The DUT sends AAC RTP media stream to ONVIF Client over RTSP

27. The DUT sends RTCP sender report to ONVIF Client.

28. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

29. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

30. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send AAC RTP media streaming to ONVIF Client.

- The DUT sent not only AAC RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.19 MEDIA STREAMING – G.711 (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-2-1-37

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

- in G711 - required audio encoding
- in UDP - Transport Protocol
- in RTP-Unicast - Stream Type
- in IPv6 - IP version
- out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in G.711 - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.20 MEDIA STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-2-1-38

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
2. Start an ONVIF Client.
3. Start the DUT.
4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters
   - in G711 - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv6 - IP version
   - out *streamUri* - Uri for media streaming
5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in G.711 - expected media stream encoding
6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.21 MEDIA STREAMING – G.711 (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-2-1-39

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP/RTSP/TCP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters
    - in G711 - required audio encoding
    - in HTTP - Transport Protocol
    - in RTP-Unicast - Stream Type
    - in IPv6 - IP version

- out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

   - in G.711 - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.22 MEDIA STREAMING – G.726 (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-2-1-40

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in G726 - required audio encoding

   - in UDP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

   - in G.726 - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.23 MEDIA STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-2-1-41

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
2. Start an ONVIF Client.
3. Start the DUT.
4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters
   - in G726 - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv6 - IP version
   - out *streamUri* - Uri for media streaming
5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters

- in *streamUri* - Uri for media streaming
- in audio - media type
- in G.726 - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.24 MEDIA STREAMING – G.726 (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-2-1-42

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP/RTSP/TCP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

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

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in G726 - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv6 - IP version
   - out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in G.726 - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.25 MEDIA STREAMING – AAC (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-2-1-43

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in AAC - required audio encoding

   - in UDP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

   - in AAC - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.26 MEDIA STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-2-1-44

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

2. Start an ONVIF Client.

3. Start the DUT.

4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

- in AAC - required audio encoding
- in HTTP - Transport Protocol
- in RTP-Unicast - Stream Type
- in IPv6 - IP version
- out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in audio - media type
   - in AAC - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.27 MEDIA STREAMING – AAC (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-2-1-45

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP/RTSP/TCP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile to have audio configuration with required audio encoder and any video configuration following the procedure mentioned in Annex A.5 with the following input and output parameters

   - in AAC - required audio encoding

   - in HTTP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in audio - media type

   - in AAC - expected media stream encoding

6. ONVIF Client restores settings of Audio Encoder Configuration and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

5.2.1.28 AUDIO STREAMING – G.711 (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-2-1-46

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/UDP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.711”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/UDP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.711 RTP media stream to ONVIF Client over UDP

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

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

- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.711 RTP media streaming to ONVIF Client.
- The DUT sent not only G.711 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.29 AUDIO STREAMING – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-2-1-47

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioSourceConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.711”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “HTTP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/RTSP/HTTP/TCP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.711 RTP media stream to ONVIF Client over HTTP.

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.711 RTP media streaming to ONVIF Client.
- The DUT sent not only G.711 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.30 AUDIO STREAMING – G.711 (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-2-1-48

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.711 media streaming based on RTP/RTSP/TCP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (Action/MaxNVTProfiles or other) execute Annex A.9.
6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.
9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.
10. DUT sends the list of audio encoder configurations compatible with the received media profile token.
11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.
12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as 'testprofileX' and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = "G.711", Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.711 encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = "RTP-Unicast", Transport.Protocol = "RTSP") to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP/RTSP/TCP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.711 RTP media stream to ONVIF Client over RTSP

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.
30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
31. The DUT sends 200 OK Response and terminates the RTSP Session.
32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.711 RTP media streaming to ONVIF Client.
- The DUT sent not only G.711 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.31 AUDIO STREAMING – G.726 (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-2-1-49

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/UDP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where

ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.726”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.
22. The DUT sends 200 OK message and SDP information.
23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/UDP.
24. The DUT sends 200 OK message and the media stream information.
25. ONVIF Client invokes RTSP PLAY request.
26. The DUT sends 200 OK message and starts media streaming.
27. The DUT sends G.726 RTP media stream to ONVIF Client over UDP
28. The DUT sends RTCP sender report to ONVIF Client.
29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.
30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
31. The DUT sends 200 OK Response and terminates the RTSP Session.
32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.

- The DUT did not send G.726 RTP media streaming to ONVIF Client.
- The DUT sent not only G.726 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.32 AUDIO STREAMING – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-2-1-50

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioSourceConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.726”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “HTTP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/RTSP/HTTP/TCP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.726 RTP media stream to ONVIF Client over HTTP.

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP media streaming to ONVIF Client.
- The DUT sent not only G.726 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.33 AUDIO STREAMING – G.726 (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-2-1-51

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify G.726 media streaming based on RTP/RTSP/TCP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G.726”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set G.726 encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP/RTSP/TCP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.726 RTP media stream to ONVIF Client over RTSP

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.
30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.
31. The DUT sends 200 OK Response and terminates the RTSP Session.
32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP media streaming to ONVIF Client.
- The DUT sent not only G.726 RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.34 AUDIO STREAMING – AAC (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-2-1-52

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

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/UDP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where

ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/UDP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends AAC RTP media stream to ONVIF Client over UDP

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.

- The DUT did not send AAC RTP media streaming to ONVIF Client.
- The DUT sent not only AAC RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.35 AUDIO STREAMING – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-2-1-53

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP, RTSP over HTTP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP-Unicast/RTSP/HTTP/TCP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioSourceConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “HTTP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Unicast/RTSP/HTTP/TCP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends AAC RTP media stream to ONVIF Client over HTTP.

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send AAC RTP media streaming to ONVIF Client.
- The DUT sent not only AAC RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.1.36 AUDIO STREAMING – AAC (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-2-1-54

Specification Coverage: RTP/RTSP/HTTP/TCP, RTP, RTCP, Stream control, RTSP

Command Under Test: None.

WSDL Reference: None

Test Purpose: To verify AAC media streaming based on RTP/RTSP/TCP Transport for IPv6 for case if there is only Audio Source Configuration and Audio encoder Configuration in Media Profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. RTP/RTSP/TCP transport protocol is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface. The DUT is configured with Manual IPv6 or LinkLocal IPv6 address.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio source configurations compatible with profile.
7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first **GetCompatibleAudioSourceConfigurationsResponse** from the **GetAudioSourceConfigurationsResponse** message.
9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.
10. DUT sends the list of audio encoder configurations compatible with the received media profile token.
11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.
12. ONVIF Client verifies the list of audio encoder configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 11-12 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 6-12 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 12.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Session Timeout = t1 and force persistence = false) to set AAC encoding.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP/RTSP/TCP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends AAC RTP media stream to ONVIF Client over RTSP

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

32. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send AAC RTP media streaming to ONVIF Client.
- The DUT sent not only AAC RTP media streaming to ONVIF Client.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.1 for Invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.2 Multicast

5.2.2.1 AUDIO ENCODER CONFIGURATION – MULTICAST PORT (IPv4)

Test Case ID: RTSS-2-2-9

Specification Coverage: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, Start multicast streaming, Stop multicast streaming

Command Under Test: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, StartMulticastStreaming, StopMulticastStreaming.

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of Multicast port for Audio Encoder Configuration in case of the same Multicast address for all Audio Encoder Configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. At least one Audio Encoder is supported by the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetAudioEncoderConfigurationsRequest** message to retrieve audio configuration list.

4. Verify the **GetAudioEncoderConfigurationsResponse** message.

5. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (ConfigurationToken = AECToken1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, where AECToken1 is first audio encoder configuration token from **GetAudioEncoderConfigurationsResponse** message) to change multicast port and address for audio encoder configuration.

6. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

7. ONVIF Client invokes **GetAudioEncoderConfigurationRequest** message (ConfigurationToken = AECToken1) to get audio encoder configuration.

8. Verify the **GetAudioEncoderConfigurationResponse** message (ConfigurationToken = Token1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

9. Repeat steps 5-8 for the rest Audio Encoder configurations supported by the DUT with using different multicast ports and the same multicast addresses for Audio Encoder Configurations.

10. Find or create media profile with Audio Source Configuration and Audio Encoder Configuration with token AECToken1 (see Annex A.10). If it is not possible skip steps 11-18 and go to the step 19.

11. Configure multicast settings for other entities from profile if required (see Annex A.11).

12. ONVIF Client invokes **StartMulticastStreamingRequest** message (ProfileToken = [profile token from the step 10]) to start multicast streaming.

13. Verify the **StartMulticastStreamingResponse** from the DUT.

14. The DUT sends audio RTP multicast media stream to multicast IPv4 address over UDP.

15. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

16. ONVIF Client validates that specified multicast address and port are used.

17. ONVIF Client invokes **StopMulticastStreamingRequest** message (ProfileToken = [profile token from the step 10]) to stop multicast streaming from specified port.

18. Verify the **StopMulticastStreamingResponse** from the DUT.

19. Repeat steps 10-18 for the rest Audio Encoder configuration supported by the DUT.

20. Restore Audio Encoder Configurations settings.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send valid **GetAudioEncoderConfigurationsResponse** message.

- The DUT did not send valid **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send valid **GetAudioEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during steps 5-6.

- The DUT did not send valid **StartMulticastStreamingResponse** message.

- The DUT did not send valid audio stream to specified multicast address and port.

- The DUT did not send valid **StopMulticastStreamingResponse** message.

- It is not possible to find or create profile for all Audio Encoder Configurations.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

5.2.2.2 AUDIO ENCODER CONFIGURATION – MULTICAST ADDRESS (IPv4)

Test Case ID: RTSS-2-2-10

Specification Coverage: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, Start multicast streaming, Stop multicast streaming

Command Under Test: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, StartMulticastStreaming, StopMulticastStreaming.

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of Multicast address for Audio Encoder Configuration in case of the same Multicast port for all Audio Encoder Configurations.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. At least one Audio Encoder is supported by the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client invokes **GetAudioEncoderConfigurationsRequest** message to retrieve audio configuration list.

4. Verify the **GetAudioEncoderConfigurationsResponse** message.

5. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (ConfigurationToken = AECToken1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, where AECToken1 is first audio encoder configuration token from **GetAudioEncoderConfigurationsResponse** message) to change multicast port and address for audio encoder configuration.

6. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

7. ONVIF Client invokes **GetAudioEncoderConfigurationRequest** message (ConfigurationToken = AECToken1) to get audio encoder configuration.

8. Verify the **GetAudioEncoderConfigurationResponse** message (ConfigurationToken = Token1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

9. Repeat steps 5-8 for the rest Audio Encoder configurations supported by the DUT with using different multicast address and the same multicast port for Audio Encoder Configurations.

10. Find or create media profile with Audio Source Configuration and Audio Encoder Configuration with token AECToken1 (see Annex A.10). If it is not possible skip steps 11-18 and go to the step 19.

11. Configure multicast settings for other entities from profile if required (see Annex A.11).

12. ONVIF Client invokes **StartMulticastStreamingRequest** message (ProfileToken = [profile token from the step 10]) to start multicast streaming.

13. Verify the **StartMulticastStreamingResponse** from the DUT.

14. The DUT sends audio RTP multicast media stream to multicast IPv4 address over UDP.

15. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

16. ONVIF Client validates that specified multicast address and port are used.

17. ONVIF Client invokes **StopMulticastStreamingRequest** message (ProfileToken = [profile token from the step 10]) to stop multicast streaming from specified port.

18. Verify the **StopMulticastStreamingResponse** from the DUT.

19. Repeat steps 10-18 for the rest Audio Encoder configuration supported by the DUT.

20. Restore Audio Encoder Configurations settings.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send valid **GetAudioEncoderConfigurationsResponse** message.
- The DUT did not send valid **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send valid **GetAudioEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during steps 5-6.
- The DUT did not send valid **StartMulticastStreamingResponse** message.
- The DUT did not send valid audio stream to specified multicast address and port.
- The DUT did not send valid **StopMulticastStreamingResponse** message.
- It is not possible to find or create profile for all Audio Encoder Configurations.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

5.2.2.3 MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-2-2-11

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, G.711 over RTP, Stream control, RTSP

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify G.711 media streaming based on RTP-Multicast/UDP Transport for IPv4.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. A media profile with G.711 audio encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

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

3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.

6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio source configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 12-13 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 7-13 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 13

15. DUT adds the audio encoder configuration to the profile and sends the response.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G711”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.711 encoding and Multicast settings.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.711 RTP multicast media stream to multicast IPv4 address over UDP.

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send G.711 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.2.4 MEDIA STREAMING – G.711 (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-2-2-12

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, G.711 over RTP, Stream control, RTSP

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify G.711 media streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT. A media profile with G.711 audio encoder configuration exists. RTP-Multicast/ UDP transport protocol is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio source configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 12-13 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.711 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.711 encoding skip this step and repeat steps 7-13 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 13

16. DUT adds the audio encoder configuration to the profile and sends the response.

17. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G711”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv6”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.711 encoding and Multicast settings.

18. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

19. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

20. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

21. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

22. ONVIF Client invokes RTSP DESCRIBE request.

23. The DUT sends 200 OK message and SDP information.

24. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP.

25. The DUT sends 200 OK message and the media stream information.

26. ONVIF Client invokes RTSP PLAY request.

27. The DUT sends 200 OK message and starts media streaming.

28. The DUT sends G.711 RTP multicast media stream to multicast IPv6 address over UDP.

29. The DUT sends RTCP sender report to ONVIF Client.

30. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

31. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

32. The DUT sends 200 OK Response and terminates the RTSP Session.

33. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.
- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **SetVideoEncoderConfigurationResponse** message.
- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.711 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.2.5 MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-2-2-13

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, G.726 over RTP, Stream control, RTSP

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify G.726 media streaming based on RTP-Multicast/UDP Transport for IPv4.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and G.726 is implemented by DUT. A media profile with G.726 audio encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. Media Service was received from the DUT

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio source configurations sent by DUT.

12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 12-13 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 7-13 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 13

15. DUT adds the audio encoder configuration to the profile and sends the response.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G726”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.726 encoding and Multicast settings.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

19. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends G.726 RTP multicast media stream to multicast IPv4 address over UDP.

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send G.726 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.2.6 MEDIA STREAMING – G.726 (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-2-2-14

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, G.726 over RTP, Stream control, RTSP

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify G.726 media streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and G.726 is implemented by DUT. A media profile with G.726 audio encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = 'testprofileX' and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = 'testprofileX') to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio source configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 12-13 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports G.726 encoding go to the next step. If there is no AudioEncoderConfiguration which supports G.726 encoding skip this step and repeat steps 7-13 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as 'testprofileX' and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 13

16. DUT adds the audio encoder configuration to the profile and sends the response.

17. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = "G726", Bitrate = r1, SampleRate = r2, Multicast.Address = ["IPv6", "multicastAddress1"], Multicast.Port = "port1", TTL = "ttl1", Session Timeout = t1 and force persistence = false) to set G.726 encoding and Multicast settings.

18. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

19. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

20. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

21. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

22. ONVIF Client invokes RTSP DESCRIBE request.

23. The DUT sends 200 OK message and SDP information.

24. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP.

25. The DUT sends 200 OK message and the media stream information.

26. ONVIF Client invokes RTSP PLAY request.

27. The DUT sends 200 OK message and starts media streaming.

28. The DUT sends G.726 RTP multicast media stream to multicast IPv6 address over UDP.

29. The DUT sends RTCP sender report to ONVIF Client.

30. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

31. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

32. The DUT sends 200 OK Response and terminates the RTSP Session.

33. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT did not send **SetVideoEncoderConfigurationResponse** message.
- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.2.7 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-2-2-15

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, AAC over RTP, Stream control, RTSP

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify AAC media streaming based on RTP-Multicast/UDP Transport for IPv4.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and AAC is implemented by DUT. A media profile with AAC audio encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. Media Service was received from the DUT

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.
9. DUT sends the list of audio encoder configurations compatible with the received media profile token.
10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.
11. ONVIF Client verifies the list of audio source configurations sent by DUT.
12. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT
13. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 12-13 for other VECs from

GetCompatibleAudioEncoderConfigurationsResponse message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 7-13 for other AudioSource.

14. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as 'testprofileX' and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 13

15. DUT adds the audio encoder configuration to the profile and sends the response.

16. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = "AAC", Bitrate = r1, SampleRate = r2, Multicast.Address = ["IPv4", "multicastAddress1"], Multicast.Port = "port1", TTL = "ttl1", Session Timeout = t1 and force persistence = false) to set AAC encoding and Multicast settings.

17. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

18. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = "RTP-Multicast", Transport.Protocol = "UDP") to retrieve media stream URI for the selected media profile.

19. The DUT sends the GetStreamUriResponse message with RTSP URI.

20. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

21. ONVIF Client invokes RTSP DESCRIBE request.

22. The DUT sends 200 OK message and SDP information.

23. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP.

24. The DUT sends 200 OK message and the media stream information.

25. ONVIF Client invokes RTSP PLAY request.

26. The DUT sends 200 OK message and starts media streaming.

27. The DUT sends AAC RTP multicast media stream to multicast IPv4 address over UDP.

28. The DUT sends RTCP sender report to ONVIF Client.

29. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

30. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

31. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **SetVideoEncoderConfigurationResponse** message.
- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send AAC RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.2.2.8 MEDIA STREAMING – AAC (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-2-2-16

Specification Coverage: RTP data transfer via UDP, RTP, RTCP, AAC over RTP, Stream control, RTSP

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify AAC media streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio is supported by DUT and AAC is implemented by DUT. A media profile with AAC audio encoder configuration exists. RTP-Multicast/UDP transport protocol is supported by DUT. Media Service was received from the DUT. IPv6 is turned on for network interface.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

5. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

6. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.

7. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.

9. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

10. DUT sends the list of audio encoder configurations compatible with the received media profile token.

11. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

12. ONVIF Client verifies the list of audio source configurations sent by DUT.

13. ONVIF Client will invoke **GetAudioEncoderConfigurationOptionsRequest** message (MediaProfile token, VECToken1, where VECToken1 is the token of the first VEC from **GetCompatibleAudioEncoderConfigurationsResponse** message) to retrieve audio encoder configuration options from the DUT

14. DUT sends **GetAudioEncoderConfigurationOptionsResponse** message. If **GetAudioEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 12-13 for other VECs from **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is AudioEncoderConfiguration which supports AAC encoding go to the next step. If there is no AudioEncoderConfiguration which supports AAC encoding skip this step and repeat steps 7-13 for other AudioSource.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = ConfigurationToken1, where ConfigurationToken1 is token of AudioEncoderConfiguration found in step 13

16. DUT adds the audio encoder configuration to the profile and sends the response.

17. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv6”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set AAC encoding and Multicast settings.

18. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

19. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Multicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

20. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

21. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

22. ONVIF Client invokes RTSP DESCRIBE request.

23. The DUT sends 200 OK message and SDP information.

24. ONVIF Client invokes RTSP SETUP request with transport parameter RTP-Multicast/UDP.

25. The DUT sends 200 OK message and the media stream information.

26. ONVIF Client invokes RTSP PLAY request.

27. The DUT sends 200 OK message and starts media streaming.

28. The DUT sends AAC RTP multicast media stream to multicast IPv6 address over UDP.

29. The DUT sends RTCP sender report to ONVIF Client.

30. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

31. ONVIF Client invokes RTSP TEARDOWN control request at the end of media streaming to terminate the RTSP session.

32. The DUT sends 200 OK Response and terminates the RTSP Session.

33. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.
- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.
- The DUT did not send **SetVideoEncoderConfigurationResponse** message.
- The DUT did not send **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send AAC RTP multicast media streaming for corresponding multicast IP.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.3 Audio & Video Streaming

5.3.1 Unicast

5.3.1.1 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP-Unicast/UDP)

Test Case ID: RTSS-3-1-10

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on RTP/UDP Unicast Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. Real-time streaming is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

- in G711 - required audio encoding
- in UDP - Transport Protocol
- in RTP-Unicast - Stream Type
- in IPv4 - IP version
- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.711 - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.2 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-3-1-11

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on HTTP Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. Real-time streaming is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G711 - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.711 - expected audio stream encoding

5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.3 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP/RTSP/TCP)

Test Case ID: RTSS-3-1-12

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on RTP/RTSP/TCP using RTSP tunnel.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. Real-time streaming is supported by the DUT. RTP/RTSP/TCP media streaming is supported by the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters
   - in G711 - required audio encoding
   - in RTSP - Transport Protocol

- in RTP-Unicast - Stream Type
- in IPv4 - IP version
- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters
    - in *streamUri* - Uri for media streaming
    - in video - 1st media type
    - in audio - 2nd media type
    - in *videoEncoding* - expected video stream encoding
    - in G.711 - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.4 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP-Unicast/UDP)

Test Case ID: RTSS-3-1-13

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on RTP/UDP Unicast Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. Real-time streaming is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters
   - in G726 - required audio encoding
   - in UDP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder
4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.726 - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.5 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-3-1-14

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on HTTP Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. Real-time streaming is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters
   - in G726 - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version

- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.726 - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.6 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP/RTSP/TCP)

Test Case ID: RTSS-3-1-15

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on RTP/RTSP/TCP using RTSP tunnel.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. Real-time streaming is supported by the DUT. RTP/RTSP/TCP media streaming is supported by the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G726 - required audio encoding

   - in RTSP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv4 - IP version

   - out *streamUri* - Uri for media streaming

   - out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in video - 1st media type

   - in audio - 2nd media type

   - in *videoEncoding* - expected video stream encoding

   - in G.726 - expected audio stream encoding

5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.7 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP-Unicast/UDP)

Test Case ID: RTSS-3-1-16

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on RTP/UDP Unicast Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. Real-time streaming is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in AAC - required audio encoding

   - in UDP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv4 - IP version

- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in AAC - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.8 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP-Unicast/RTSP/HTTP/TCP)

Test Case ID: RTSS-3-1-17

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on HTTP Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. Real-time streaming is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in AAC - required audio encoding
   - in HTTP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in AAC - expected audio stream encoding

5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.9 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP/RTSP/TCP)

Test Case ID: RTSS-3-1-18

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on RTP/RTSP/TCP using RTSP tunnel.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. Real-time streaming is supported by the DUT. RTP/RTSP/TCP media streaming is supported by the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters
   - in AAC - required audio encoding
   - in RTSP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming

- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in AAC - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.10 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-3-1-19

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on RTP/UDP Unicast Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. Real-time streaming is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

    - in G711 - required audio encoding

    - in UDP - Transport Protocol

    - in RTP-Unicast - Stream Type

    - in IPv6 - IP version

    - out *streamUri* - Uri for media streaming

    - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

    - in *streamUri* - Uri for media streaming

    - in video - 1st media type

    - in audio - 2nd media type

    - in *videoEncoding* - expected video stream encoding

    - in G.711 - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.11 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-3-1-20

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on HTTP Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. Real-time streaming is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

- in G711 - required audio encoding
- in HTTP - Transport Protocol
- in RTP-Unicast - Stream Type
- in IPv6 - IP version
- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters
    - in *streamUri* - Uri for media streaming
    - in video - 1st media type
    - in audio - 2nd media type
    - in *videoEncoding* - expected video stream encoding
    - in G.711 - expected audio stream encoding
6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.12 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-3-1-21

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on RTP/RTSP/TCP using RTSP tunnel for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. Real-time streaming is supported by the DUT. RTP/RTSP/TCP media streaming is supported by the DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G711 - required audio encoding

   - in RTSP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

   - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in video - 1st media type

   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.711 - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.13 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-3-1-22

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on RTP/UDP Unicast Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. Real-time streaming is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G726 - required audio encoding
   - in UDP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv6 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.726 - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.14 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-3-1-23

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on HTTP Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. Real-time streaming is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G726 - required audio encoding

   - in HTTP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

- out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.726 - expected audio stream encoding
6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.15 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-3-1-24

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on RTP/RTSP/TCP using RTSP tunnel for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. Real-time streaming is supported by the DUT. RTP/RTSP/TCP media streaming is supported by the DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.
4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters
   - in G726 - required audio encoding
   - in RTSP - Transport Protocol
   - in RTP-Unicast - Stream Type
   - in IPv6 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder
5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.726 - expected audio stream encoding
6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.16 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP-Unicast/UDP, IPv6)

Test Case ID: RTSS-3-1-25

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on RTP/UDP Unicast Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. Real-time streaming is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

- in AAC - required audio encoding
- in UDP - Transport Protocol
- in RTP-Unicast - Stream Type
- in IPv6 - IP version
- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/UDP by following the procedure mentioned in Annex A.15 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in AAC - expected audio stream encoding
6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.17 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP-Unicast/RTSP/HTTP/TCP, IPv6)

Test Case ID: RTSS-3-1-26

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on HTTP Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. Real-time streaming is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in AAC - required audio encoding

   - in HTTP - Transport Protocol

   - in RTP-Unicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

   - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/HTTP by following the procedure mentioned in Annex A.17 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in video - 1st media type

   - in audio - 2nd media type

- in *videoEncoding* - expected video stream encoding
- in AAC - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.1.18 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP/RTSP/TCP, IPv6)

Test Case ID: RTSS-3-1-27

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on RTP/RTSP/TCP using RTSP tunnel for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. Real-time streaming is supported by the DUT. RTP/RTSP/TCP media streaming is supported by the DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

    - in AAC - required audio encoding
    - in RTSP - Transport Protocol
    - in RTP-Unicast - Stream Type
    - in IPv6 - IP version
    - out *streamUri* - Uri for media streaming
    - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Unicast/RTSP by following the procedure mentioned in Annex A.18 with the following input and output parameters

    - in *streamUri* - Uri for media streaming
    - in video - 1st media type
    - in audio - 2nd media type
    - in *videoEncoding* - expected video stream encoding
    - in AAC - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2 Multicast

5.3.2.1 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-3-2-16

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on RTP-Multicast/UDP Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. RTP-Multicast/UDP transport protocol is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G711 - required audio encoding
   - in UDP - Transport Protocol
   - in RTP-Multicast - Stream Type
   - in IPv4 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Multicast/UDP by following the procedure mentioned in Annex A.16 with the following input and output parameters

    - in *streamUri* - Uri for media streaming
    - in video - 1st media type
    - in audio - 2nd media type
    - in *videoEncoding* - expected video stream encoding
    - in G.711 - expected audio stream encoding

5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2.2 MEDIA STREAMING – VIDEO AND G.711 AUDIO (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-3-2-17

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.711 Audio and Video streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.711 encoding is supported by DUT. RTP-Multicast/UDP transport protocol is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

    - in G711 - required audio encoding

    - in UDP - Transport Protocol

    - in RTP-Multicast - Stream Type

    - in IPv6 - IP version

    - out *streamUri* - Uri for media streaming

    - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Multicast/UDP by following the procedure mentioned in Annex A.16 with the following input and output parameters

    - in *streamUri* - Uri for media streaming

    - in video - 1st media type

    - in audio - 2nd media type

    - in *videoEncoding* - expected video stream encoding

    - in G.711 - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2.3 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-3-2-18

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on RTP-Multicast/UDP Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. RTP-Multicast/UDP transport protocol is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

    - in G726 - required audio encoding

    - in UDP - Transport Protocol

    - in RTP-Multicast - Stream Type

    - in IPv4 - IP version

- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Multicast/UDP by following the procedure mentioned in Annex A.16 with the following input and output parameters
   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in G.726 - expected audio stream encoding
5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2.4 MEDIA STREAMING – VIDEO AND G.726 AUDIO (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-3-2-19

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify G.726 Audio and Video streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. G.726 encoding is supported by DUT. RTP-Multicast/UDP transport protocol is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in G726 - required audio encoding

   - in UDP - Transport Protocol

   - in RTP-Multicast - Stream Type

   - in IPv6 - IP version

   - out *streamUri* - Uri for media streaming

   - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Multicast/UDP by following the procedure mentioned in Annex A.16 with the following input and output parameters

   - in *streamUri* - Uri for media streaming

   - in video - 1st media type

   - in audio - 2nd media type

   - in *videoEncoding* - expected video stream encoding

   - in G.726 - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.

7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2.5 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP-Multicast/UDP, IPv4)

Test Case ID: RTSS-3-2-20

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on RTP-Multicast/UDP Transport.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. RTP-Multicast/UDP transport protocol is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

    - in AAC - required audio encoding

- in UDP - Transport Protocol
- in RTP-Multicast - Stream Type
- in IPv4 - IP version
- out *streamUri* - Uri for media streaming
- out *videoEncoding* - video encoder

4. ONVIF Client tries to start and decode media streaming over RTP-Multicast/UDP by following the procedure mentioned in Annex A.16 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding
   - in AAC - expected audio stream encoding

5. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 3.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2.6 MEDIA STREAMING – VIDEO AND AAC AUDIO (RTP-Multicast/UDP, IPv6)

Test Case ID: RTSS-3-2-21

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

Command Under Test: None

WSDL Reference: None

Test Purpose: To verify AAC Audio and Video streaming based on RTP-Multicast/UDP Transport for IPv6.

Pre-Requisite: Media Service is received from the DUT. Audio streaming is supported by DUT. AAC encoding is supported by DUT. RTP-Multicast/UDP transport protocol is supported by DUT. IPv6 is supported by DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client turns on IPv6 network interface by following the procedure mentioned in Annex A.23.

4. ONVIF Client configures a media profile and retrieves a stream uri for video and audio streaming by following the procedure mentioned in Annex A.6 with the following input and output parameters

   - in AAC - required audio encoding
   - in UDP - Transport Protocol
   - in RTP-Multicast - Stream Type
   - in IPv6 - IP version
   - out *streamUri* - Uri for media streaming
   - out *videoEncoding* - video encoder

5. ONVIF Client tries to start and decode media streaming over RTP-Multicast/UDP by following the procedure mentioned in Annex A.16 with the following input and output parameters

   - in *streamUri* - Uri for media streaming
   - in video - 1st media type
   - in audio - 2nd media type
   - in *videoEncoding* - expected video stream encoding

- in AAC - expected audio stream encoding

6. ONVIF Client restores settings of Video Encoder Configuration, Audio Encoder Configuration, and Media Profile changed at step 4.
7. ONVIF Client will restore the original settings by following the procedure mentioned in Annex A.24.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT does not pass all assertions.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.3.2.7 VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT PORTS

Test Case ID: RTSS-3-2-22

Specification Coverage: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, Start multicast streaming, Stop multicast streaming

Command Under Test: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of Multicast port for Audio Encoder Configuration in case of the same Multicast addresses for Audio Encoder Configuration and Video Encoder Configuration in the same profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. At least one Audio Encoder is supported by the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a Media Profile with video configurations and with G.711 audio configurations by following the procedure mentioned in Annex A.20 with the following input and output parameters

   - in G.711 - required audio encoding

   - out *profile* - Media Profile with Video Source Configuration, Video Encoder Configuration, Audio Source Configuration, and Audio Encoder Configuration with the required audio encoding

4. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message message (ProfileToken = *profile*.@token).

5. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.

6. ONVIF Client invokes **GetAudioEncoderConfigurationOptionsRequest** message message (ProfileToken = *profile*.@token).

7. Verify the **GetAudioEncoderConfigurationOptionsResponse** message from the DUT.

8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = *profile*.VideoEncoderConfiguration.Encoding, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set Multicast settings. These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse**.

9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

10. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G711”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port2”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.711 encoding and Multicast settings. These values will be taken from the **GetAudioEncoderConfigurationOptionsResponse**.

11. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

12. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message to get audio encoder configuration.

13. Verify the **GetVideoEncoderConfigurationResponse** message (Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

14. ONVIF Client invokes **GetAudioEncoderConfigurationRequest** message (ConfigurationToken = AECToken1) to get audio encoder configuration.

15. Verify the **GetAudioEncoderConfigurationResponse** message (Multicast.Address = ["IPv4", "multicastAddress1"], Multicast.Port = "port2") from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

16. ONVIF Client removes Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

    - in {MetadataConfiguration} - list of configuration types to remove from Media Profile

    - in *profile* - Media Profile to update

17. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.

18. Verify the **StartMulticastStreamingResponse** message from the DUT.

19. The DUT sends video and G.711 RTP multicast media stream to multicast IPv4 address over UDP.

20. The DUT sends RTCP sender report to ONVIF Client.

21. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

22. ONVIF Client validates that specified multicast address and port are used.

23. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming

24. Verify the **StopMulticastStreamingResponse** message from the DUT.

25. Verify that multicast stream is stopped by the DUT.

26. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send valid **GetVideoEncoderConfigurationsResponse** message.

- The DUT did not send valid **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send valid **GetVideoEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during step 8.
- The DUT did not send valid **GetAudioEncoderConfigurationsResponse** message.
- The DUT did not send valid **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send valid **GetAudioEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during step 10.
- The DUT did not send valid **StartMulticastStreamingResponse** message.
- The DUT did not send valid audio and video stream to specified multicast address and port.
- The DUT did not send valid **StopMulticastStreamingResponse** message.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

5.3.2.8 VIDEO AND AUDIO ENCODER CONFIGURATION – DIFFERENT ADDRESS

Test Case ID: RTSS-3-2-23

Specification Coverage: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, Start multicast streaming, Stop multicast streaming

Command Under Test: SetAudioEncoderConfiguration, GetAudioEncoderConfiguration, SetVideoEncoderConfiguration, GetVideoEncoderConfiguration, StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify changing and applying of Multicast port for Audio Encoder Configuration in case of the same Multicast ports for Audio Encoder Configuration and Video Encoder Configuration in the same profile.

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Media Service was received from the DUT. At least one Audio Encoder is supported by the DUT. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a Media Profile with video configurations and with G.711 audio configurations by following the procedure mentioned in Annex A.20 with the following input and output parameters

   - in G.711 - required audio encoding

   - out *profile* - Media Profile with Video Source Configuration, Video Encoder Configuration, Audio Source Configuration, and Audio Encoder Configuration with the required audio encoding

4. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message message (ProfileToken = *profile*.@token).

5. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT.

6. ONVIF Client invokes **GetAudioEncoderConfigurationOptionsRequest** message message (ProfileToken = *profile*.@token).

7. Verify the **GetAudioEncoderConfigurationOptionsResponse** message from the DUT.

8. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = *profile*.VideoEncoderConfiguration.Encoding, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings. These values will be taken from the **GetVideoEncoderConfigurationOptionsResponse**.

9. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

10. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G711”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress2”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.711 encoding and Multicast settings. These values will be taken from the GetAudioEncoderConfigurationOptionsResponse.

11. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

12. ONVIF Client invokes **GetVideoEncoderConfigurationRequest** message to get audio encoder configuration.

13. Verify the **GetVideoEncoderConfigurationResponse** message (Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

14. ONVIF Client invokes **GetAudioEncoderConfigurationRequest** message (ConfigurationToken = AECToken1) to get audio encoder configuration.

15. Verify the **GetAudioEncoderConfigurationResponse** message (Multicast.Address = ["IPv4", "multicastAddress2"], Multicast.Port = "port1") from the DUT. Check that new setting for Multicast.Port and Multicast.Address was applied.

16. ONVIF Client removes Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

    - in {*MetadataConfiguration*} - list of configuration types to remove from Media Profile

    - in *profile* - Media Profile to update

17. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.

18. Verify the **StartMulticastStreamingResponse** message from the DUT.

19. The DUT sends video and G.711 RTP multicast media stream to multicast IPv4 address over UDP.

20. The DUT sends RTCP sender report to ONVIF Client.

21. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

22. ONVIF Client validates that specified multicast address and port are used.

23. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming

24. Verify the **StopMulticastStreamingResponse** message from the DUT.

25. Verify that multicast stream is stopped by the DUT.

26. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send valid **GetVideoEncoderConfigurationsResponse** message.

- The DUT did not send valid **SetVideoEncoderConfigurationResponse** message.
- The DUT did not send valid **GetVideoEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during step 8.
- The DUT did not send valid **GetAudioEncoderConfigurationsResponse** message.
- The DUT did not send valid **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send valid **GetAudioEncoderConfigurationResponse** message with new Multicast.Address and Multicast.Port values that were set during step 10.
- The DUT did not send valid **StartMulticastStreamingResponse** message.
- The DUT did not send valid audio and video stream to specified multicast address and port.
- The DUT did not send valid **StopMulticastStreamingResponse** message.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

5.4 Notification Streaming Interface

5.4.1 NOTIFICATION STREAMING

Test Case ID: RTSS-4-1-3

Specification Coverage: CreateProfile, GetVideoSourceConfigurations, GetMetadataConfigurations, AddVideoSourceConfiguration, AddMetadataConfiguration, SetMetadataConfiguration, GetStreamUri, SetSynchronizationPoint, DeleteProfile, Notification Streaming Interface.

Command Under Test: CreateProfile, GetVideoSourceConfigurations, GetMetadataConfigurations, AddVideoSourceConfiguration, AddMetadataConfiguration, SetMetadataConfiguration, GetStreamUri, SetSynchronizationPoint, DeleteProfile.

WSDL Reference: media.wsdl

Test Purpose: To verify Notification Streaming

Pre-Requisite: The device needs to provide at least one topic representing a certain property. If the device does not support a property event the vendor SHALL make sure that another event is sent during testing. Real-time streaming supported by the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfile** (name = “Test”) to create a new empty profile that is used for this test scenario.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (Action/MaxNVTProfiles or other) execute Annex A.9.
5. ONVIF Client will invoke **GetMetadataConfigurations** to retrieve all existing MetadataConfigurations of the device.
6. Verify that the DUT sends a valid **GetMetadataConfigurationResponse** (that contains at least one MetadataConfiguration).
7. ONVIF Client will invoke **GetVideoSourceConfigurations**.
8. Verify that the DUT sends a valid **GetVideoSourceConfigurationResponse**.
9. ONVIF Client will select a VideoSourceConfiguration and add this configuration to the created profile.
10. Verify that the DUT sends a valid **AddVideoSourceConfigurationResponse**.
11. ONVIF Client will select a MetadataConfiguration and add this configuration to the created profile.
12. Verify that the DUT sends a valid **AddMetadataConfigurationResponse**.
13. ONVIF Client will invoke **SetMetadataConfiguration**(Analytics = false, Events is empty) to configure the Metadata stream; The ONVIF Client is interested in receiving all events, therefore no Filter is applied. For details on the usage of the MetadataConfiguration elements, see Annex A.7.
14. Verify that the DUT sends a valid **SetMetadataConfigurationResponse**.
15. ONVIF Client will invoke **GetStreamUri** (ProfileToken, RTP-Unicast).
16. Verify that the DUT sends a **GetStreamUriResponse** including a valid StreamUri.
17. ONVIF Client will invoke RTSP Describe to retrieve the sdp file.
18. Verify that DUT sends a 200 OK Response.
19. Validate sdp file (sdp file contains only one media section; rtpmap = vnd.onvif.metadata).
20. ONVIF Client will invoke RTSP Setup for the Metadata stream.

21. Verify that the DUT send a 200 OK Response.

22. ONVIF Client will invoke RTSP Play.

23. Verify that the DUT sends a 200 OK Response.

24. Receive and validate RTP Notification messages.

25. ONVIF Client will invoke the **SetSynchronizationPoint** command to trigger events; if the device does not support property events the vendor SHALL trigger the events manually.

26. Validate that DUT sends a valid **SetSynchronizationPointResponse**.

27. Verify that at least one RTP Notification is sent.

28. Receive and validate RTP Notification messages and check that the PropertyOperation is ”Initialized” or “Changed” if it is an Property event.

29. ONVIF Client will invoke RTSP Teardown to terminate the RTSP session.

30. Verify that DUT sends a 200 OK Response.

31. If used created Media Profile, then ONVIF Client invokes **DeleteProfile** request. Otherwise ONVIF client skip rest steps and restore profile settings.

32. The DUT deletes the media profile and sends the response.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send a **CreateProfileResponse**

- The DUT did not send a valid **CreateProfileResponse**

- The DUT did not send valid **GetMetadataConfigurationsResponse**; At least one MetadataConfiguration SHALL be present

- The DUT did not send a valid **GetVideoSourceConfigurationsResponse**

- The DUT did not send a valid **AddVideoSourceConfigurationResponse**

- The DUT did not send a valid **AddMetadataConfigurationResponse**

- The DUT did not send a valid **SetMetadataConfigurationResponse**

- The DUT did not send a **GetStreamUriResponse** including a valid StreamUri

- The DUT did not send a 200 OK RTSP DESCRIBE Response

- The DUT did not send a valid sdp file

- The DUT did not send a 200 OK RTSP SETUP Response

- The DUT did not send a 200 OK RTSP PLAY Response

- The DUT did not send RTP data

- The DUT did not send a **SetSynchronizationPointResponse**

- The DUT did not send at least one event

- The DUT did not send RTP data with PropertyOperation = ”Initialized” or “Changed” if it is an property event

- The DUT did not send a 200 OK RTSP TEARDOWN Response

- The DUT did not send a **DeleteProfileResponse**

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: If profile was deleted during Annex A.9 execution, ONVIF Client restores the deleted profile and profile settings.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.5 Start And Stop Multicast Streaming

5.5.1 START AND STOP MULTICAST STREAMING – JPEG (IPv4)

Test Case ID: RTSS-5-1-7

Specification Coverage: Start multicast streaming, Stop multicast streaming

Command Under Test: StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify the behavior of StartMulticastStreaming command and StopMulticastStreaming in case of Video Streaming (JPEG).

Pre-Requisite: Video Multicasting is supported by DUT. Real-time streaming supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. A media profile with video source configuration and JPEG video encoder configuration. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with JPEG video encoding support by following the procedure mentioned in Annex A.4.

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = “JPEG”, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set JPEG encoding and Multicast settings.

5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

6. ONVIF Client removes Audio Source Configuration, Audio Encoder Configuration and Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters

    - in { *AudioEncoderConfiguration, AudioSourceConfiguration, MetadataConfiguration*} - list of configuration types to remove from Media Profile

    - in *profile* - Media Profile to update

7. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.

8. Verify the **StartMulticastStreamingResponse** message from the DUT.

9. The DUT sends JPEG RTP multicast media stream to multicast IPv4 address over UDP.

10. The DUT sends RTCP sender report to ONVIF Client.

11. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

12. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming.

13. Verify the **StopMulticastStreamingResponse** message from the DUT.

14. Verify that multicast stream is stopped by the DUT.

15. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetVideoEncoderConfigurationResponse**.

- The DUT sent invalid **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send **SetAudioEncoderConfigurationResponse**.

- The DUT sent invalid **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send **SetMetadataConfigurationResponse**.

- The DUT sent invalid **SetMetadataConfigurationResponse** message.

- The DUT did not send **StartMulticastStreamingResponse** message.

- The DUT sent invalid **StartMulticastStreamingResponse** message.

- The DUT did not send **StopMulticastStreamingResponse** message.

- The DUT sent invalid **StopMulticastStreamingResponse** message.

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send JPEG RTP multicast media streaming for corresponding multicast IP.

- The DUT did not stop multicast streaming after **StopMulticastStreamingResponse**.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

5.5.2 START AND STOP MULTICAST STREAMING – VIDEO AND G.711 AUDIO (IPv4)

Test Case ID: RTSS-5-1-9

Specification Coverage: Start multicast streaming, Stop multicast streaming

Command Under Test: StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify the behavior of StartMulticastStreaming command and StopMulticastStreaming in case of Video and Audio Streaming.

Pre-Requisite: Video Multicasting is supported by DUT. Real-time streaming supported by DUT. Audio is supported by DUT. Audio Multicasting is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. The DUT is configured without rotation (either 0 degrees or rotation is OFF) if rotation is supported.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a Media Profile with video configurations and with G.711 audio configurations by following the procedure mentioned in Annex A.20 with the following input and output parameters

   - in G.711 - required audio encoding

   - out *profile* - Media Profile with Video Source Configuration, Video Encoder Configuration, Audio Source Configuration, and Audio Encoder Configuration with the required audio encoding

4. ONVIF Client invokes **SetVideoEncoderConfigurationRequest** message (Encoding = *profile*.VideoEncoderConfiguration.Encoding, Resolution = [“Width”, “Height”], Quality = q1, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set Multicast settings.

5. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.
6. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G711”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.711 encoding and Multicast settings.
7. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.
8. ONVIF Client removes Metadata Configuration from the Media Profile that will be used for multicast streaming by following the procedure mentioned in by following the procedure mentioned in Annex A.29 with the following input and output parameters
   - in {*MetadataConfiguration*} - list of configuration types to remove from Media Profile
   - in *profile* - Media Profile to update
9. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.
10. Verify the **StartMulticastStreamingResponse** message from the DUT.
11. The DUT sends video and G.711 audio RTP multicast media stream to multicast IPv4 address over UDP.
12. The DUT sends RTCP sender report to ONVIF Client.
13. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.
14. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming.
15. Verify the **StopMulticastStreamingResponse** message from the DUT.
16. Verify that multicast stream is stopped by the DUT.
17. The DTT restores changed profiles and configurations.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetVideoEncoderConfigurationResponse**.

- The DUT sent invalid **SetVideoEncoderConfigurationResponse** message.

- The DUT did not send **SetAudioEncoderConfigurationResponse**.

- The DUT sent invalid **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send **SetMetadataConfigurationResponse**.

- The DUT sent invalid **SetMetadataConfigurationResponse** message.

- The DUT did not send **StartMulticastStreamingResponse** message.

- The DUT sent invalid **StartMulticastStreamingResponse** message.

- The DUT did not send **StopMulticastStreamingResponse** message.

- The DUT sent invalid **StopMulticastStreamingResponse** message.

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send video and audio RTP multicast media streaming for corresponding multicast IP.

- The DUT did not stop multicast streaming after **StopMulticastStreamingResponse**.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: See Annex A.27 and Annex A.28 for Resolution and Frame Rate selection logic description.

5.5.3 START AND STOP MULTICAST STREAMING – G.711 (IPv4, ONLY AUDIO PROFILE)

Test Case ID: RTSS-5-1-10

Specification Coverage: Start multicast streaming, Stop multicast streaming

Command Under Test: StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify the behavior of StartMulticastStreaming command and StopMulticastStreaming in case of Audio Streaming (G.711).

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT.Audio Multicasting is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = 'testprofileX'.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.
7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = 'testprofileX' and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.
8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = 'testprofileX') to retrieve the list of audio encoder configurations compatible with profile.
9. DUT sends the list of audio encoder configurations compatible with the received media profile token.
10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.
11. ONVIF Client verifies the list of audio source configurations sent by DUT.
12. ONVIF Client invokes **GetAudioEncoderConfigurationOptionsRequest** message (ProfileToken = TestProfileX, ConfigurationToken = AECToken1), where AECToken1 is the token of the first AudioEncoderConfiguration from the **GetCompatibleAudioEncoderConfigurationsResponse** message.
13. Verify **GetAudioEncoderConfigurationOptionsResponse** from the DUT. Check supported Audio Codec. If G.711 is supported skip step 14.

14. Repeat steps 12-13 for others Audio Encoder Configurations from the **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is no any others AudioEncoderConfiguration repeat steps 7-14 for other AudioSource from **GetCompatibleAudioSourceConfigurationsResponse** message.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = AECTokenX, where AECTokenX is token of AudioEncoderConfiguration found in the step 13.

16. DUT adds the audio encoder configuration to the profile and sends the response.

17. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G711”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.711 encoding and Multicast settings.

18. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

19. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.

20. Verify the **StartMulticastStreamingResponse** message from the DUT.

21. The DUT sends G.711 RTP multicast media stream to multicast IPv4 address over UDP.

22. The DUT sends RTCP sender report to ONVIF Client.

23. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming.

25. Verify the **StopMulticastStreamingResponse** message from the DUT.

26. Verify that multicast stream is stopped by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse**.

- The DUT did not send AudioEncoderConfiguration which supports G.711 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT sent invalid **SetAudioEncoderConfigurationResponse** message.
- The DUT did not send **StartMulticastStreamingResponse** message.
- The DUT sent invalid **StartMulticastStreamingResponse** message.
- The DUT did not send **StopMulticastStreamingResponse** message.
- The DUT sent invalid **StopMulticastStreamingResponse** message.
- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.711 RTP multicast media streaming for corresponding multicast IP.
- The DUT did not stop multicast streaming after **StopMulticastStreamingResponse**.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.5.4 START AND STOP MULTICAST STREAMING – G.726 (IPv4, ONLY AUDIO PROFILE)

Test Case ID: RTSS-5-1-11

Specification Coverage: Start multicast streaming, Stop multicast streaming

Command Under Test: StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify the behavior of StartMulticastStreaming command and StopMulticastStreaming in case of Audio Streaming (G.726).

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio Multicasting is supported by DUT and G.726 is implemented by DUT. Media Service was received from the DUT

Test Configuration: ONVIF Client and DUT

Test Procedure:

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

3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.

4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (**Action/MaxNVTProfiles** or other) execute Annex A.9.

5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.

6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio source configurations sent by DUT.

12. ONVIF Client invokes **GetAudioEncoderConfigurationOptionsRequest** message (ProfileToken = TestProfileX, ConfigurationToken = AECToken1), where AECToken1 is the token of the first AudioEncoderConfiguration from the **GetCompatibleAudioEncoderConfigurationsResponse** message.

13. Verify **GetAudioEncoderConfigurationOptionsResponse** from the DUT. Check supported Audio Codec. If G.726 is supported skip step 14.

14. Repeat steps 12-13 for others Audio Encoder Configurations from the **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is no any others AudioEncoderConfiguration repeat steps 7-14 for other AudioSource from **GetCompatibleAudioSourceConfigurationsResponse** message.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = AECTokenX, where AECTokenX is token of AudioEncoderConfiguration found in the step 13.

16. DUT adds the audio encoder configuration to the profile and sends the response.

17. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “G726”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set G.726 encoding and Multicast settings.

18. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

19. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.

20. Verify the **StartMulticastStreamingResponse** message from the DUT.

21. The DUT sends G.726 RTP multicast media stream to multicast IPv4 address over UDP.

22. The DUT sends RTCP sender report to ONVIF Client.

23. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming.

25. Verify the **StopMulticastStreamingResponse** message from the DUT.

26. Verify that multicast stream is stopped by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse**.

- The DUT did not send AudioEncoderConfiguration which supports G.726 encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT sent invalid **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send **StartMulticastStreamingResponse** message.

- The DUT sent invalid **StartMulticastStreamingResponse** message.

- The DUT did not send **StopMulticastStreamingResponse** message.

- The DUT sent invalid **StopMulticastStreamingResponse** message.

- The DUT did not send valid RTP header in one or more media streams.
- The DUT did not send RTCP sender report correctly.
- The DUT did not send G.726 RTP multicast media streaming for corresponding multicast IP.
- The DUT did not stop multicast streaming after **StopMulticastStreamingResponse**.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.5.5 START AND STOP MULTICAST STREAMING – AAC (IPv4, ONLY AUDIO PROFILE)

Test Case ID: RTSS-5-1-12

Specification Coverage: Start multicast streaming, Stop multicast streaming

Command Under Test: StartMulticastStreaming, StopMulticastStreaming

WSDL Reference: media.wsdl

Test Purpose: To verify the behavior of StartMulticastStreaming command and StopMulticastStreaming in case of Audio Streaming (AAC).

Pre-Requisite: Media is supported by DUT. Real-time streaming supported by DUT. Audio Multicasting is supported by DUT and AAC is implemented by DUT. Media Service was received from the DUT

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **CreateProfileRequest** message with ProfileToken = ‘testprofileX’.
4. Verify **CreateProfileResponse** message from the DUT. If the DUT sends a SOAP 1.2 fault message (Action/MaxNVTProfiles or other) execute Annex A.9.
5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurationsRequest** message to retrieve all existing audio source configurations of the DUT.
6. Verify **GetCompatibleAudioSourceConfigurationsResponse** message.

7. ONVIF Client invokes **AddAudioSourceConfigurationResponse** message with ProfileToken = ‘testprofileX’ and ConfigurationToken = ConfigurationToken1 where ConfigurationToken1 is the first ConfigurationToken from the **GetCompatibleAudioSourceConfigurationsResponse** message.

8. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = ‘testprofileX’) to retrieve the list of audio encoder configurations compatible with profile.

9. DUT sends the list of audio encoder configurations compatible with the received media profile token.

10. ONVIF Client verifies the **GetCompatibleAudioEncoderConfigurationsResponse** message.

11. ONVIF Client verifies the list of audio source configurations sent by DUT.

12. ONVIF Client invokes **GetAudioEncoderConfigurationOptionsRequest** message (ProfileToken = TestProfileX, ConfigurationToken = AECToken1), where AECToken1 is the token of the first AudioEncoderConfiguration from the **GetCompatibleAudioEncoderConfigurationsResponse** message.

13. Verify **GetAudioEncoderConfigurationOptionsResponse** from the DUT. Check supported Audio Codec. If AAC is supported skip step 14.

14. Repeat steps 12-13 for others Audio Encoder Configurations from the **GetCompatibleAudioEncoderConfigurationsResponse** message. If there is no any others AudioEncoderConfiguration repeat steps 7-14 for other AudioSource from **GetCompatibleAudioSourceConfigurationsResponse** message.

15. ONVIF Client invokes **AddAudioEncoderConfigurationRequest** message with ProfileToken as ‘testprofileX’ and ConfigurationToken = AECTokenX, where AECTokenX is token of AudioEncoderConfiguration found in the step 13.

16. DUT adds the audio encoder configuration to the profile and sends the response.

17. ONVIF Client invokes **SetAudioEncoderConfigurationRequest** message (Encoding = “AAC”, Bitrate = r1, SampleRate = r2, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”, TTL = “ttl1”, Session Timeout = t1 and force persistence = false) to set AAC encoding and Multicast settings.

18. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

19. ONVIF Client will invoke **StartMulticastStreamingRequest** message (ProfileToken) to start multicast streaming.

20. Verify the **StartMulticastStreamingResponse** message from the DUT.

21. The DUT sends AAC RTP multicast media stream to multicast IPv4 address over UDP.

22. The DUT sends RTCP sender report to ONVIF Client.

23. ONVIF Client validates the received RTP and RTCP packets, decodes and renders them.

24. ONVIF Client will invoke **StopMulticastStreamingRequest** message (ProfileToken) to stop multicast streaming.

25. Verify the **StopMulticastStreamingResponse** message from the DUT.

26. Verify that multicast stream is stopped by the DUT.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **SetAudioEncoderConfigurationResponse**.

- The DUT did not send AudioEncoderConfiguration which supports AAC encoding in the **GetCompatibleAudioEncoderConfigurationsResponse** message.

- The DUT sent invalid **SetAudioEncoderConfigurationResponse** message.

- The DUT did not send **StartMulticastStreamingResponse** message.

- The DUT sent invalid **StartMulticastStreamingResponse** message.

- The DUT did not send **StopMulticastStreamingResponse** message.

- The DUT sent invalid **StopMulticastStreamingResponse** message.

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

- The DUT did not send RTCP sender report correctly.

- The DUT did not send AAC RTP multicast media streaming for corresponding multicast IP.

- The DUT did not stop multicast streaming after **StopMulticastStreamingResponse**.

Note: See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

5.6 Audio Backchannel Streaming

5.6.1 BACKCHANNEL – G.711 (RTP-Unicast/UDP, IPv4)

Test Case ID: RTSS-6-1-1

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for G.711 audio streaming using RTP-Unicast/UDP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and G.711 Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with G.711 decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.
4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.
5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.
6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.
7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

10. The DUT sends 200 OK message and SDP information.

11. Verify SDP information from the DUT. Check that DUT return audio backchannel and G.711 on the list of supported codecs for backchannel.

12. ONVIF Client invokes RTSP SETUP request for backchannel with transport parameter RTP-Unicast/UDP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

13. The DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message.

16. ONVIF Client sends RTP Unicast G.711 audio stream to DUT over UDP.

17. The DUT processes a stream and send it to Output.

18. The DUT sends RTCP sender report to ONVIF Client.

19. ONVIF Client validates the received RTCP packets.

20. ONVIF Client invokes RTSP TEARDOWN control request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

21. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send **SetAudioOutputConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

- The DUT did not send valid SDP information:
- There is no backchannel (marked with a=sendonly attribute)
- There is no G.711 on the list of supported codecs for backchannel
- The DUT did not send RTCP sender report correctly.
- The DUT did not process G.711 RTP media streaming.
- The DUT did not send G.711 RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.2 BACKCHANNEL – G.711 (RTP/RTSP/TCP, IPv4)

Test Case ID: RTSS-6-1-3

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for G.711 audio streaming using RTP/RTSP/TCP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and G.711 Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with G.711 decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.

4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.

5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.

6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

10. The DUT sends 200 OK message and SDP information.

11. Verify SDP information from the DUT. Check that DUT return audio backchannel and G.711 on the list of supported codecs for backchannel.

12. ONVIF Client invokes RTSP SETUP request for backchannel with transport parameter RTP/RTSP/TCP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

13. The DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message.

16. ONVIF Client sends RTP Unicast G.711 audio stream to DUT over UDP.

17. The DUT processes a stream and send it to Output.

18. The DUT sends RTCP sender report to ONVIF Client.

19. ONVIF Client validates the received RTCP packets.

20. ONVIF Client invokes RTSP TEARDOWN control request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

21. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send **SetAudioOutputConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

- The DUT did not send valid SDP information:

- There is no backchannel (marked with a=sendonly attribute)

- There is no G.711 on the list of supported codecs for backchannel

- The DUT did not send RTCP sender report correctly.

- The DUT did not process G.711 RTP media streaming.

- The DUT did not send G.711 RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.3 BACKCHANNEL – G.726 (RTP-Unicast/UDP, IPv4)

Test Case ID: RTSS-6-1-4

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for G.726 audio streaming using RTP-Unicast/UDP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and G.726 Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with G.726 decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.

4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.

5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.

6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

10. The DUT sends 200 OK message and SDP information.

11. Verify SDP information from the DUT. Check that DUT return audio backchannel and G.726 on the list of supported codecs for backchannel.

12. ONVIF Client invokes RTSP SETUP request for backchannel with transport parameter RTP-Unicast/UDP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

13. The DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message.

16. ONVIF Client sends RTP Unicast G.726 audio stream to DUT over UDP.

17. The DUT processes a stream and send it to Output.

18. The DUT sends RTCP sender report to ONVIF Client.

19. ONVIF Client validates the received RTCP packets.

20. ONVIF Client invokes RTSP TEARDOWN control request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

21. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send **SetAudioOutputConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

- The DUT did not send valid SDP information:

- There is no backchannel (marked with a=sendonly attribute)

- There is no G.726 on the list of supported codecs for backchannel

- The DUT did not send RTCP sender report correctly.

- The DUT did not process G.726 RTP media streaming.

- The DUT did not send G.726 RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.4 BACKCHANNEL – G.726 (RTP/RTSP/TCP, IPv4)

Test Case ID: RTSS-6-1-6

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for G.726 audio streaming using RTP/RTSP/TCP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and G.726 Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with G.726 decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.
4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.
5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.
6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.
7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).
10. The DUT sends 200 OK message and SDP information.
11. Verify SDP information from the DUT. Check that DUT return audio backchannel and G.726 on the list of supported codecs for backchannel.
12. ONVIF Client invokes RTSP SETUP request for backchannel with transport parameter RTP/RTSP/TCP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).
13. The DUT sends 200 OK message and the media stream information.
14. ONVIF Client invokes RTSP PLAY request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).
15. The DUT sends 200 OK message.
16. ONVIF Client sends RTP Unicast G.726 audio stream to DUT over UDP.
17. The DUT processes a stream and send it to Output.
18. The DUT sends RTCP sender report to ONVIF Client.

19. ONVIF Client validates the received RTCP packets.

20. ONVIF Client invokes RTSP TEARDOWN control request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

21. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.
- The DUT did not send **SetAudioOutputConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid SDP information:
- There is no backchannel (marked with a=sendonly attribute)
- There is no G.726 on the list of supported codecs for backchannel
- The DUT did not send RTCP sender report correctly.
- The DUT did not process G.726 RTP media streaming.
- The DUT did not send G.726 RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.5 BACKCHANNEL – AAC (RTP-Unicast/UDP, IPv4)

Test Case ID: RTSS-6-1-7

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for AAC audio streaming using RTP-Unicast/UDP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and AAC Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with AAC decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.

4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.

5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.

6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “UDP”) to retrieve media stream URI for the selected media profile.

7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

10. The DUT sends 200 OK message and SDP information.

11. Verify SDP information from the DUT. Check that DUT return audio backchannel and AAC on the list of supported codecs for backchannel.

12. ONVIF Client invokes RTSP SETUP request for backchannel with transport parameter RTP-Unicast/UDP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

13. The DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message.

16. ONVIF Client sends RTP Unicast AAC audio stream to DUT over UDP.

17. The DUT processes a stream and send it to Output.

18. The DUT sends RTCP sender report to ONVIF Client.

19. ONVIF Client validates the received RTCP packets.

20. ONVIF Client invokes RTSP TEARDOWN control request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

21. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send **SetAudioOutputConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid SDP information:
- There is no backchannel (marked with a=sendonly attribute)
- There is no AAC on the list of supported codecs for backchannel
- The DUT did not send RTCP sender report correctly.
- The DUT did not process AAC RTP media streaming.
- The DUT did not send AAC RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.6 BACKCHANNEL – AAC (RTP/RTSP/TCP, IPv4)

Test Case ID: RTSS-6-1-9

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for AAC audio streaming using RTP/RTSP/TCP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and AAC Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with AAC decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.

4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.

5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.

6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

10. The DUT sends 200 OK message and SDP information.

11. Verify SDP information from the DUT. Check that DUT return audio backchannel and AAC on the list of supported codecs for backchannel.

12. ONVIF Client invokes RTSP SETUP request for backchannel with transport parameter RTP/RTSP/TCP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

13. The DUT sends 200 OK message and the media stream information.

14. ONVIF Client invokes RTSP PLAY request with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message.

16. ONVIF Client sends RTP Unicast AAC audio stream to DUT over UDP.

17. The DUT processes a stream and send it to Output.

18. The DUT sends RTCP sender report to ONVIF Client.

19. ONVIF Client validates the received RTCP packets.

20. ONVIF Client invokes RTSP TEARDOWN control request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

21. The DUT sends 200 OK Response and terminates the RTSP Session.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.
- The DUT did not send **SetAudioOutputConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid SDP information:
- There is no backchannel (marked with a=sendonly attribute)
- There is no AAC on the list of supported codecs for backchannel
- The DUT did not send RTCP sender report correctly.
- The DUT did not process AAC RTP media streaming.
- The DUT did not send AAC RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.7 BACKCHANNEL STREAMING BY POST – G.711 (RTP-Unicast/RTSP/HTTP/TCP, IPv4)

Test Case ID: RTSS-6-1-10

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for G.711 audio streaming using RTP/RTSP/TCP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and G.711 Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with G.711 decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.
4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.
5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.
6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.
7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

11. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) on HTTP POST connection.

12. The DUT sends 200 OK message and SDP information on HTTP GET connection.

13. Verify SDP information from the DUT. Check that DUT return audio backchannel and G.711 on the list of supported codecs for backchannel.

14. ONVIF Client invokes RTSP SETUP request on HTTP POST connection for backchannel with transport parameter RTP-Unicast/RTSP/HTTP/TCP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message and the media stream information on HTTP GET connection.

16. ONVIF Client invokes RTSP PLAY request on HTTP POST connection with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

17. The DUT sends 200 OK message on HTTP GET connection.

18. ONVIF Client sends RTP Unicast G.711 audio stream to DUT on HTTP POST connection.

19. The DUT processes a stream and send it to Output.

20. The DUT sends RTCP sender report to ONVIF Client.

21. ONVIF Client validates the received RTCP packets.

22. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

23. The DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.
- The DUT did not send **SetAudioOutputConfigurationResponse** message.
- The DUT did not send **GetStreamUriResponse** message.
- The DUT did not send valid **GetStreamUriResponse** message.
- The DUT did not send RTSP 200 OK response for RTSP DESCRIBE, SETUP, PLAY and TEARDOWN requests.
- The DUT did not send valid SDP information:
- There is no backchannel (marked with a=sendonly attribute)
- There is no G.711 on the list of supported codecs for backchannel
- The DUT did not send RTCP sender report correctly.
- The DUT did not process G.711 RTP media streaming.
- The DUT did not send G.711 RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.8 BACKCHANNEL STREAMING BY POST – G.726 (RTP-Unicast/RTSP/HTTP/TCP, IPv4)

Test Case ID: RTSS-6-1-11

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for G.726 audio streaming using RTP/RTSP/TCP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and G.726 Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client selects a media profile with G.726 decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.
4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.
5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.
6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.
7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.
8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.
9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.
10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.
11. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) on HTTP POST connection.
12. The DUT sends 200 OK message and SDP information on HTTP GET connection.
13. Verify SDP information from the DUT. Check that DUT return audio backchannel and G.726 on the list of supported codecs for backchannel.

14. ONVIF Client invokes RTSP SETUP request on HTTP POST connection for backchannel with transport parameter RTP-Unicast/RTSP/HTTP/TCP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message and the media stream information on HTTP GET connection.

16. ONVIF Client invokes RTSP PLAY request on HTTP POST connection with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

17. The DUT sends 200 OK message on HTTP GET connection.

18. ONVIF Client sends RTP Unicast G.726 audio stream to DUT on HTTP POST connection.

19. The DUT processes a stream and send it to Output.

20. The DUT sends RTCP sender report to ONVIF Client.

21. ONVIF Client validates the received RTCP packets.

22. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

23. The DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send **SetAudioOutputConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

- The DUT did not send valid SDP information:

- There is no backchannel (marked with a=sendonly attribute)
- There is no G.726 on the list of supported codecs for backchannel
- The DUT did not send RTCP sender report correctly.
- The DUT did not process G.726 RTP media streaming.
- The DUT did not send G.726 RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

5.6.9 BACKCHANNEL STREAMING BY POST – AAC (RTP-Unicast/RTSP/HTTP/TCP, IPv4)

Test Case ID: RTSS-6-1-12

Specification Coverage: Back Channel Connection, RTSP Require- Tag, Connection setup for a bi-directional connection

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Backchannel for AAC audio streaming using RTP/RTSP/TCP transport for IPv4.

Pre-Requisite: Media is supported by the DUT. Audio backchannel is supported by the DUT and AAC Decoder is implemented by DUT. Media Service entry point is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

3. ONVIF Client selects a media profile with AAC decoding and not only www.onvif.org/ver20/HalfDuplex/Client send primacy support by following the procedure mentioned in Annex A.25.

4. ONVIF Client invokes **SetAudioOutputConfigurationRequest** message (SendPrimacy = “www.onvif.org/ver20/HalfDuplex/Client”, OutputLevel = o1 and force persistence = false) to set output configuration for backchannel.

5. Verify the **SetAudioOutputConfigurationResponse** message from the DUT.

6. ONVIF Client invokes **GetStreamUriRequest** message (ProfileToken, Stream = “RTP-Unicast”, Transport.Protocol = “RTSP”) to retrieve media stream URI for the selected media profile.

7. The DUT sends the **GetStreamUriResponse** message with RTSP URI.

8. ONVIF Client verifies the RTSP media stream URI provided by the DUT.

9. ONVIF Client invokes HTTP GET Request on DUT and establishes DUT to ONVIF Client connection for RTP data transfer.

10. ONVIF Client invokes HTTP POST Request and establishes ONVIF Client to DUT connection for RTSP control requests.

11. ONVIF Client invokes RTSP DESCRIBE request with additional Require-tag (Require: www.onvif.org/ver20/backchannel) on HTTP POST connection.

12. The DUT sends 200 OK message and SDP information on HTTP GET connection.

13. Verify SDP information from the DUT. Check that DUT return audio backchannel and AAC on the list of supported codecs for backchannel.

14. ONVIF Client invokes RTSP SETUP request on HTTP POST connection for backchannel with transport parameter RTP-Unicast/RTSP/HTTP/TCP, with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

15. The DUT sends 200 OK message and the media stream information on HTTP GET connection.

16. ONVIF Client invokes RTSP PLAY request on HTTP POST connection with additional Require-tag (Require: www.onvif.org/ver20/backchannel).

17. The DUT sends 200 OK message on HTTP GET connection.

18. ONVIF Client sends RTP Unicast AAC audio stream to DUT on HTTP POST connection.

19. The DUT processes a stream and send it to Output.

20. The DUT sends RTCP sender report to ONVIF Client.

21. ONVIF Client validates the received RTCP packets.

22. ONVIF Client invokes RTSP TEARDOWN control request on HTTP POST connection with additional Require-tag (Require: www.onvif.org/ver20/backchannel) at the end of media streaming to terminate the RTSP session.

23. The DUT sends 200 OK Response on HTTP GET connection and closes the HTTP GET connection.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not have valid media profile.

- The DUT did not send **SetAudioOutputConfigurationResponse** message.

- The DUT did not send **GetStreamUriResponse** message.

- The DUT did not send valid **GetStreamUriResponse** message.

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

- The DUT did not send valid SDP information:

- There is no backchannel (marked with a=sendonly attribute)

- There is no AAC on the list of supported codecs for backchannel

- The DUT did not send RTCP sender report correctly.

- The DUT did not process AAC RTP media streaming.

- The DUT did not send AAC RTP media streaming to Output from media profile.

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests. See Annex A.1 for Invalid RTP header definition.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: ONVIF Clients checks IP type of Stream URI by following the procedure mentioned in Annex A.30.

Note: ONVIF Clients checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31.

Annex A Helper Procedures and Additional Notes

A.1 Invalid RTP Header

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

A.2 StreamSetup syntax for GetStreamUri

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

1. RTP Unicast over UDP
2. RTP over RTSP over HTTP over TCP
3. RTP over RTSP over TCP

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

1. RTP Unicast over UDP

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

2. RTP over RTSP over HTTP over TCP

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

3. RTP over RTSP over TCP

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

A.3 I-frame insertion time interval

‘I-frame insertion time interval’ is the time interval between two consecutive I-frames sent by DUT.

ONVIF Client calculates this value by using the ‘GovLength’ parameter in the Video encoder configuration. ONVIF Client has to configure ‘GovLength’ to a large enough value so that there will be a sufficient time difference between two I-frames.

For SetSynchronizationPoint test cases in the “Real Time Streaming” section, ONVIF Client follows this procedure to verify that I-frame is inserted as a result of SetSynchronizationPoint request.

ONVIF Client waits for an I-frame before invoking SetSynchronizationPoint command.

After receiving I-frame, ONVIF Client starts a timer with time out period less than ‘I-frame insertion time interval’ and immediately invokes SetSynchronizationPoint command.

ONVIF Client waits for the I-frame and verifies that it receives I-frame before the timeout period.

A.4 Media Profile Configuration for Video Streaming

For the execution of real time streaming - video test cases, ONVIF Client has to select and configure the media profile based on the required video codec.

ONVIF Client follows the following procedure to configure the media profile.

Retrieve media profiles by invoking GetProfiles command.

Retrieve supported video encoder configuration options for a media profile by invoking GetVideoEncoderConfigurationOptions (media profile token) command. Check whether the selected media profile supports the required video codec.

Repeat test procedure-2 for all media profiles till a media profile with the required video codec support is found.

A.5 Device Configuration for Audio Streaming with Media Profile

Name: HelperDeviceConfigurationForMediaAudioStreaming

Procedure Purpose: Helper procedure to configure Media profile with Video and Audio Configurations for audio streaming, and get stream URI from the DUT.

Pre-requisite: Media Service is received from the DUT. Audio is supported by the DUT.

Input: Required audio encoding (*requiredAudioEncoding*), Transport protocol *protocol*, Stream Type (*streamType*), IP version (*ipVersion*) (optional, default = IPv4) .

Returns: Stream Uri (*streamUri*).

Procedure:

1. ONVIF Client selects a Media Profile with video configurations and with required audio configurations by following the procedure mentioned in Annex A.20 with the following input and output parameters

   - in *requiredAudioEncoding* - required audio encoding

   - out *profile* - Media Profile with Video Source Configuration, Video Encoder Configuration, Audio Source Configuration, and Audio Encoder Configuration with the required audio encoding

2. ONVIF Client invokes **SetAudioEncoderConfiguration** request with parameters

   - Configuration.@token := *aec*.@token

   - Configuration.Name := *aec*.Name

   - Configuration.UseCount := *aec*.UseCount

   - Configuration.Encoding := *aec*.Encoding

   - Configuration.Bitrate := *aec*.Bitrate

- Configuration.SampleRate := *aec*.SampleRate
- if *streamType* != RTP-Multicast:
  - Configuration.Multicast := *profile*.VideoEncoderConfiguration.Multicast
- if *streamType* = RTP-Multicast and *ipVersion* = IPv4:
  - Configuration.Multicast.Address.Type := IPv4
  - Configuration.Multicast.Address.IPv4Address := multicast IPv4 address
  - Configuration.Multicast.Address.IPv6Address skipped
  - Configuration.Multicast.Port := port for multicast streaming
  - Configuration.Multicast.TTL := 1
  - Configuration.Multicast.AutoStart := false
- if *streamType* = RTP-Multicast and *ipVersion* = IPv6:
  - Configuration.Multicast.Address.Type := IPv6
  - Configuration.Multicast.Address.IPv4Address skipped
  - Configuration.Multicast.Address.IPv6Address := multicast IPv6 address
  - Configuration.Multicast.Port := port for multicast streaming
  - Configuration.Multicast.TTL := 1
  - Configuration.Multicast.AutoStart := false
- Configuration.SessionTimeout := *aec*.SessionTimeout
- ForcePersistence := false

3. The DUT responds with **SetAudioEncoderConfigurationResponse** message.

4. ONVIF Client retrieves a stream uri for Media Profile for required transport protocol by following the procedure mentioned in Annex A.21 with the following input and output parameters

   - in *streamType* - Stream Type
   - in *protocol* - Transport protocol

- in *profile*.@token - Media profile token
- out *uri* - Stream URI

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **SetAudioEncoderConfigurationResponse** message.

A.6 Device Configuration for Video and Audio Streaming

Name: HelperDeviceConfigurationForVideoAndAudioStreaming

Procedure Purpose: Helper procedure to configure Media profile, Video Encoder Configuration, Audio Encoder Configuration, and get stream URI from the DUT for video and audio streaming.

Pre-requisite: Media Service is received from the DUT. Audio is supported by the DUT.

Input: Required audio encoding (*requiredAudioEncoding*), Transport protocol *protocol*, Stream Type (*streamType*), IP version (*ipVersion*) (optional, default = IPv4) .

Returns: Stream Uri (*streamUri*), Video Encoder set in configuread profile *videoEncoding*.

Procedure:

1. ONVIF Client selects a Media Profile with video configurations and with required audio configurations by following the procedure mentioned in Annex A.20 with the following input and output parameters
   - in *requiredAudioEncoding* - required audio encoding
   - out *profile* - Media Profile with Video Source Configuration, Video Encoder Configuration, Audio Source Configuration, and Audio Encoder Configuration with the required audio encoding
2. Set *videoEncoding* := *profile*.VideoEncoderConfiguration.Encoding.
3. ONVIF Client invokes **GetVideoEncoderConfigurationOptions** request with parameters
   - ConfigurationToken skipped
   - ProfileToken := *profile*.@token

4. DUT responds with **GetVideoEncoderConfigurationOptionsResponse** message with parameters

    - Options list =: *optionsList*

5. Set *vecOptions* := item with Encoding = *profile*.VideoEncoderConfiguration.Encoding from *optionsList* list.

6. Set *vecExtOptions* := Extension item with Encoding = *profile*.VideoEncoderConfiguration.Encoding from *optionsList* list.

7. ONVIF Client invokes **SetVideoEncoderConfiguration** request with parameters

    - Configuration.@token := *profile*.VideoSourceConfiguration.@token

    - Configuration.Name := *profile*.VideoSourceConfiguration.Name

    - Configuration.UseCount := *profile*.VideoSourceConfiguration.UseCount

    - Configuration.Encoding := *profile*.VideoSourceConfiguration.Encoding

    - Configuration.Resolution := resolution closest to 640x480 from *vecOptions*.ResolutionsAvailable list

    - Configuration.Quality := *vecOptions*.QualityRange.Min

    - if *vecExtOptions* null and *profile*.VideoEncoderConfiguration.RateControl skipped:

        - Configuration.RateControl skipped

      Otherwise:

        - Configuration.RateControl.FrameRateLimit := value closest to 25 but greater than 1 from *vecOptions*.FrameRateRange

        - Configuration.RateControl.EncodingInterval := *vecOptions*.EncodingIntervalRange.Min

        - Configuration.RateControl.BitrateLimit := min {max {*profile*.VideoEncoderConfiguration.RateControl.BitrateLimit, *vecExtOptions*.BitrateRange.Min}, *vecExtOptions*.BitrateRange.Max}

    - if *streamType* != RTP-Multicast:

        - Configuration.Multicast := *profile*.VideoEncoderConfiguration.Multicast

    - if *streamType* = RTP-Multicast and *ipVersion* = IPv4:

  - Configuration.Multicast.Address.Type := IPv4
  - Configuration.Multicast.Address.IPv4Address := multicast IPv4 address
  - Configuration.Multicast.Address.IPv6Address skipped
  - Configuration.Multicast.Port := port for multicast streaming
  - Configuration.Multicast.TTL := 1
  - Configuration.Multicast.AutoStart := false
- if *streamType* = RTP-Multicast and *ipVersion* = IPv6:
  - Configuration.Multicast.Address.Type := IPv6
  - Configuration.Multicast.Address.IPv4Address skipped
  - Configuration.Multicast.Address.IPv6Address := multicast IPv6 address
  - Configuration.Multicast.Port := port for multicast streaming
  - Configuration.Multicast.TTL := 1
  - Configuration.Multicast.AutoStart := false
- Configuration.SessionTimeout := *profile*.VideoEncoderConfiguration.SessionTimeout
- ForcePersistence := false

8. The DUT responds with **SetVideoEncoderConfigurationResponse** message.

9. ONVIF Client invokes **SetAudioEncoderConfiguration** request with parameters

   - Configuration.@token := *aec*.@token
   - Configuration.Name := *aec*.Name
   - Configuration.UseCount := *aec*.UseCount
   - Configuration.Encoding := *aec*.Encoding
   - Configuration.Bitrate := *aec*.Bitrate
   - Configuration.SampleRate := *aec*.SampleRate
   - if *streamType* != RTP-Multicast:

  - Configuration.Multicast := *profile*.VideoEncoderConfiguration.Multicast

- if *streamType* = RTP-Multicast and *ipVersion* = IPv4:

  - Configuration.Multicast.Address.Type := IPv4

  - Configuration.Multicast.Address.IPv4Address := multicast IPv4 address

  - Configuration.Multicast.Address.IPv6Address skipped

  - Configuration.Multicast.Port := port for multicast streaming

  - Configuration.Multicast.TTL := 1

  - Configuration.Multicast.AutoStart := false

- if *streamType* = RTP-Multicast and *ipVersion* = IPv6:

  - Configuration.Multicast.Address.Type := IPv6

  - Configuration.Multicast.Address.IPv4Address skipped

  - Configuration.Multicast.Address.IPv6Address := multicast IPv6 address

  - Configuration.Multicast.Port := port for multicast streaming

  - Configuration.Multicast.TTL := 1

  - Configuration.Multicast.AutoStart := false

- Configuration.SessionTimeout := *aec*.SessionTimeout

- ForcePersistence := false

10. The DUT responds with **SetAudioEncoderConfigurationResponse** message.

11. ONVIF Client retrieves a stream uri for Media Profile for required transport protocol by following the procedure mentioned in Annex A.21 with the following input and output parameters

    - in *streamType* - Stream Type

    - in *protocol* - Transport protocol

    - in *profile*.@token - Media profile token

    - out *uri* - Stream URI

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetVideoEncoderConfigurationOptionsResponse** message.
- DUT did not send **SetVideoEncoderConfigurationResponse** message.
- DUT did not send **SetAudioEncoderConfigurationResponse** message.

A.7 MetadataConfiguration for receiving / not receiving events metadata

When receiving metadata an ONVIF Client might be interested in receiving all, none or some of the events produced by DUT. The basic rules for configuring a MetadataConfiguration to achieve this are:

- To get all events: Include the Events element but do not include a filter.
- To get no events: Do not include the Events element.
- To get only some events: Include the Events element and include a filter in the element.

This means that if an Events element is not included in a MetadataConfiguration, then no Events metadata shall be included in the metadata stream for that configuration. Similarly, if an "<Events>" tag without any sub-tags is included in a MetadataConfiguration it means that all available Events metadata shall be included in the metadata stream for that configuration.

Example:

The following SetMetadataConfigurationRequest can be used when the ONVIF Client is interested in a metadata stream that includes all Events, but nothing else. The PTZStatus element is not included in the configuration, so no PTZ metadata will be included in the metadata stream. The Events element is included, but without any filter, so all Events will be included in the metadata stream.

```


<SetMetadataConfiguration>
  <Configuration token = "Test">
```

```
    <Name>TestName</Name>
    <UseCount>1</UseCount>
    <Events/>
    <Analytics>false</Analytics>
  </Configuration>
  <ForcePersistence>true</ForcePersistence>
</SetMetadataConfiguration>
```

A.8 Multicast specific field in SDP (RTSP DESCRIBE response)

In any case that multicast real-time streaming is involved in conjunction with RTSP, c = field should be included to clearly indicate the multicast address being used for streaming. Following is the example of how the c = field should be formed in the SDP (RTSP DESCRIBE response).

c = IN IP4 224.10.10.100

A.9 Create Empty Profile

Name: HelperCreateEmptyProfile

Procedure Purpose: Helper procedure to create or configure an empty media profile.

Pre-requisite: Media Service is received from the DUT.

Input: None

Returns: Empty media profile (*profile*).

Procedure:

1. ONVIF Client will invoke **GetProfiles** request message to retrieve complete profiles list.

2. Verify the **GetProfilesResponse** message from the DUT.

3. If there is profile with fixed attribute equal to false, then follows the following procedure:

    - ONVIF Client will invoke **DeleteProfile** request message (ProfileToken = Token1, where Token1 is the first ProfileToken from the **GetProfilesResponse** with fixed = false) to delete Profile.

- Verify the **DeleteProfileResponse** message from the DUT.
- Skip other steps.

4. If there are no profiles with fixed = false remove all configurations from one fixed profile and use this profile for test. If there are no profiles skip other steps and fail test.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetProfilesResponse** message.
- DUT did not send **DeleteProfileResponse** message.

A.10 Find or Create Media Profile Containing Specified Audio Encoder Configuration

For the execution of test cases with Audio configuration, ONVIF Client has to select or create the media profile contains specified Audio Encoder Configuration.

ONVIF Client follows the following procedure to select or create the media profile with specified audio encoder configuration:

1. ONVIF Client will invoke **GetProfilesRequest** message to retrieve complete profiles list.
2. Verify the **GetProfilesResponse** message from the DUT.
3. Try to find profile that contains Audio Encoder Configuration with specified token and Audio Source Configuration. If there is no such Profiles go to the next step, otherwise use one of profiles that fit to the requirements and skip other steps.
4. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.
5. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) or SOAP 1.2 fault message (**Action/MaxNVTProfiles**) from the DUT. If **CreateProfileResponse** message was received go to the step 10.
6. ONVIF Client will invoke **DeleteProfileRequest** message (ProfileToken = “Profile2”, where “Profile2” is token of profile with fixed = ”false”) to remove profile. If there are no profiles

with fixed = ”false” remove all configurations from one fixed profile, skip steps 6-9 and use this profile as profile with ProfileToken = “ProfileToken1”. If there are no profiles skip other steps and fail test.

7. Verify the **DeleteProfileResponse** message from the DUT.

8. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.

9. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) from the DUT.

10. ONVIF Client will invoke **GetCompatibleAudioSourceConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible audio source configurations list.

11. Verify the **GetCompatibleAudioSourceConfigurationsResponse** message from the DUT. If **GetCompatibleAudioSourceConfigurationsResponse** message contains empty list skip other steps (this will means that it is not possible to fined or create profile for specified audio encoder configuration).

12. ONVIF Client will invoke **AddAudioSourceConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “ASCToken1”, where “ASCToken1” is audio source configuration from **GetCompatibleAudioSourceConfigurationsResponse** message) to add audio source configuration to profile.

13. Verify the **AddAudioSourceConfigurationResponse** message from the DUT.

14. ONVIF Client will invoke **GetCompatibleAudioEncoderConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible audio encoder configurations list.

15. Verify the **GetCompatibleAudioEncoderConfigurationsResponse** message from the DUT. If **GetCompatibleAudioEncoderConfigurationsResponse** message does not contains specified audio encoder configuration repeat steps 12-15 for other audio source configuration from **GetCompatibleAudioSourceConfigurationsResponse** message. If there is no audio source configuration that was not used in steps 12-15, skip other steps (this will means that it is not possible to fined or create profile for specified audio encoder configuration).

16. ONVIF Client will invoke **AddAudioEncoderConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “AECToken1”, where “AECToken1” is audio encoder configuration from GetCompatibleAudioEncoderConfigurationsResponse message) to add audio encoder configuration to profile. Use this profile as result of procedure.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

A.11 Configure Multicast Settings for Configurations that are Included in Profile

It is necessary to have Configuration with valid Multicast settings in Profile to use StartMulticastStreaming command. As pre-requisite ONVIF Client already has received profile from the DUT.

ONVIF Client follows the following procedure to set valid Multicast parameters:

1. If profile contains VideoEncoderConfiguration (token = “VECToken1”) verify that it has invalid multicast settings, otherwise skip steps 2-3 and go to the step 4.

2. ONVIF Client will invoke **SetVideoEncoderConfigurationRequest** message (token = “VECToken1”, Multicast.Address = [“IPv4”, “multicastAddress1”], Multicast.Port = “port1”) to change multicast settings for configuration.

3. Verify the **SetVideoEncoderConfigurationResponse** message from the DUT.

4. If profile contains AudioEncoderConfiguration (token = “AECToken1”) verify that it has invalid multicast settings, otherwise skip steps 5-6 and go to the step 7.

5. ONVIF Client will invoke **SetAudioEncoderConfigurationRequest** message (token = “AECToken1”, Multicast.Address = [“IPv4”, “multicastAddress2”], Multicast.Port = “port2”) to change multicast settings for configuration.

6. Verify the **SetAudioEncoderConfigurationResponse** message from the DUT.

7. If profile contains MetadataConfiguration (token = “MCToken1”) verify that it has invalid multicast settings, otherwise skip step 8-9 and finish procedure.

8. ONVIF Client will invoke **SetMetadataConfigurationRequest** message (token = “MCToken1”, Multicast.Address = [“IPv4”, “multicastAddress3”], Multicast.Port = “port3”) to change multicast settings for configuration.

9. Verify the **SetMetadataConfigurationResponse** message from the DUT.

A.12 Find or Create Media Profile Containing Specified Video Encoder Configuration

For the execution of test cases with Video configuration, ONVIF Client has to select or create the media profile contains specified Video Encoder Configuration.

ONVIF Client follows the following procedure to select or create the media profile with specified video encoder configuration:

1. ONVIF Client will invoke **GetProfilesRequest** message to retrieve complete profiles list.
2. Verify the **GetProfilesResponse** message from the DUT.
3. Try to find profile that contains Video Encoder Configuration with specified token and Video Source Configuration. If there is no such Profiles go to the next step, otherwise use one of profiles that fit to the requirements and skip other steps.
4. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.
5. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) or SOAP 1.2 fault message (**Action/MaxNVTProfiles**) from the DUT. If **CreateProfileResponse** message was received go to the step 10.
6. ONVIF Client will invoke **DeleteProfileRequest** message (ProfileToken = “Profile2”, where “Profile2” is token of profile with fixed = ”false”) to remove profile. If there are no profiles with fixed = ”false” remove all configurations from one fixed profile, skip steps 6-9 and use this profile as profile with ProfileToken = “ProfileToken1”. If there are no profiles skip other steps and fail test.
7. Verify the **DeleteProfileResponse** message from the DUT.
8. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.
9. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) from the DUT.
10. ONVIF Client will invoke **GetCompatibleVideoSourceConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible video source configurations list.
11. Verify the **GetCompatibleVideoSourceConfigurationsResponse** message from the DUT. If GetCompatibleVideoSourceConfigurationsResponse message contains empty list skip other steps (this will means that it is not possible to fined or create profile for specified video encoder configuration).
12. ONVIF Client will invoke **AddVideoSourceConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “VSCToken1”, where “VSCToken1” is video source configuration from **GetCompatibleVideoSourceConfigurationsResponse** message) to add video source configuration to profile.
13. Verify the **AddVideoSourceConfigurationResponse** message from the DUT.
14. ONVIF Client will invoke **GetCompatibleVideoEncoderConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible video encoder configurations list.

15. Verify the **GetCompatibleVideoEncoderConfigurationsResponse** message from the DUT. If **GetCompatibleVideoEncoderConfigurationsResponse** message does not contains specified video encoder configuration repeat steps 12-15 for other video source configuration from **GetCompatibleVideoSourceConfigurationsResponse** message. If there is no video source configuration that was not used in steps 12-15, skip other steps (this will means that it is not possible to fined or create profile for specified video encoder configuration).
16. ONVIF Client will invoke **AddVideoEncoderConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “VECToken1”, where “VECToken1” is video encoder configuration from **GetCompatibleVideoEncoderConfigurationsResponse** message) to add video encoder configuration to profile. Use this profile as result of procedure.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

A.13 Find or Create Media Profile Containing Specified Video Encoder Configuration with Possibility to Set Specified Encoding

For the execution of test cases with Video configuration, ONVIF Client has to select or create the media profile contains specified Video Encoder Configuration with possibility to set specified encoding.

ONVIF Client follows the following procedure to select or create the media profile with specified video encoder configuration with possibility to set specified encoding:

1. ONVIF Client will invoke **GetProfilesRequest** message to retrieve complete profiles list.
2. Verify the **GetProfilesResponse** message from the DUT.
3. Try to find profile that contains Video Encoder Configuration with specified token and Video Source Configuration. If there is no such Profiles go to the step 6.
4. Try to invoke **GetVideoEncoderConfigurationOptionsRequest** message for first media profile found on step 3.
5. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT. If **GetVideoEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 4-5 for other profiles found on step 3. Otherwise use this profile as result of this procedure and skip other steps.
6. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.

7. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) or SOAP 1.2 fault message (**Action/MaxNVTProfiles**) from the DUT. If **CreateProfileResponse** message was received go to the step 12.

8. ONVIF Client will invoke **DeleteProfileRequest** message (ProfileToken = “Profile2”, where “Profile2” is token of profile with fixed = ”false”) to remove profile. If there are no profiles with fixed = ”false” skip other steps (this will means that it is not possible to fined or create profile for specified audio encoder configuration).

9. Verify the **DeleteProfileResponse** message from the DUT.

10. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.

11. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) from the DUT.

12. ONVIF Client will invoke **GetCompatibleVideoSourceConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible video source configurations list.

13. Verify the **GetCompatibleVideoSourceConfigurationsResponse** message from the DUT. If **GetCompatibleVideoSourceConfigurationsResponse** message contains empty list skip other steps (this will means that it is not possible to fined or create profile for specified video encoder configuration).

14. ONVIF Client will invoke **AddVideoSourceConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “VSCToken1”, where “VSCToken1” is video source configuration from **GetCompatibleVideoSourceConfigurationsResponse** message) to add video source configuration to profile.

15. Verify the **AddVideoSourceConfigurationResponse** message from the DUT.

16. ONVIF Client will invoke **GetCompatibleVideoEncoderConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible video encoder configurations list.

17. Verify the **GetCompatibleVideoEncoderConfigurationsResponse** message from the DUT. If **GetCompatibleVideoEncoderConfigurationsResponse** message does not contains specified video encoder configuration repeat steps 14-17 for other video source configuration from **GetCompatibleVideoSourceConfigurationsResponse** message. If there is no video source configuration that was not used in steps 14-17, skip other steps (this will means that it is not possible to fined or create profile for specified video encoder configuration).

18. ONVIF Client will invoke **AddVideoEncoderConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “VECToken1”, where “VECToken1”

is video encoder configuration from GetCompatibleVideoEncoderConfigurationsResponse message) to add video encoder configuration to profile.

19. ONVIF Client invokes **GetVideoEncoderConfigurationOptionsRequest** message (ProfileToken = “ProfileToken1”) to get video encoder configuration options.

20. Verify the **GetVideoEncoderConfigurationOptionsResponse** message from the DUT. If **GetVideoEncoderConfigurationOptionsResponse** message does not contains specified video encoding repeat steps 14-20 for other video source configuration from **GetCompatibleVideoSourceConfigurationsResponse** message (previously remove video encoder configuration from the profile). If there is no video source configuration that was not used in steps 14-20, skip other steps (this will means that it is not possible to fined or create profile for specified video encoder configuration).

21. Use this profile as result of procedure.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

A.14 Find or Create Specified Number of Profiles that Contains Specified Video Source Configuration and different Video Encoder Configurations

For the execution of test cases with Video configuration, ONVIF Client has to select or create the media profile contains specified Video Encoder Source.

ONVIF Client follows the following procedure to select or create the media profile with specified video encoder configuration:

1. ONVIF Client will invoke **GetProfilesRequest** message to retrieve complete profiles list.

2. Verify the **GetProfilesResponse** message from the DUT.

3. Try to find profile that contains Video Source Configuration with specified token and different Video Encoder Configuration. If number of existing profiles less than required go to the next step, otherwise limit this profiles list to requested and use them as result list.

4. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.

5. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) or SOAP 1.2 fault message (**Action/MaxNVTProfiles**) from the DUT. If **CreateProfileResponse** message was received go to the step 10.

6. ONVIF Client will invoke **DeleteProfileRequest** message (ProfileToken = “Profile2”, where “Profile2” is token of profile with fixed = ”false” and without specified video source

configuration and without video encoder configuration) to remove profile. If there are no such profiles skip other steps (this will means that it is not possible to find or create profile for specified requirements).

7. Verify the **DeleteProfileResponse** message from the DUT.

8. ONVIF Client will invoke **CreateProfileRequest** message (Name = “TestProfile1”) to create new profile.

9. Verify the **CreateProfileResponse** message (token = “ProfileToken1”, fixed = ”false”) from the DUT.

10. ONVIF Client will invoke **GetCompatibleVideoSourceConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible video source configurations list.

11. Verify the **GetCompatibleVideoSourceConfigurationsResponse** message from the DUT. If **GetCompatibleVideoSourceConfigurationsResponse** message does not contains required video source configuration skip other steps (this will means that it is not possible to find or create profile for specified requirements).

12. ONVIF Client will invoke **AddVideoSourceConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “VSCToken1”, where “VSCToken1” is video source configuration from **GetCompatibleVideoSourceConfigurationsResponse** message) to add video source configuration to profile.

13. Verify the **AddVideoSourceConfigurationResponse** message from the DUT.

14. ONVIF Client will invoke **GetCompatibleVideoEncoderConfigurationsRequest** message (ProfileToken = “ProfileToken1”) to retrieve compatible video encoder configurations list.

15. Verify the **GetCompatibleVideoEncoderConfigurationsResponse** message from the DUT. If **GetCompatibleVideoEncoderConfigurationsResponse** message contains empty list, skip other steps (this will means that it is not possible to fined or create profile for specified requirements).

16. ONVIF Client will invoke **AddVideoEncoderConfigurationRequest** message (ProfileToken = “ProfileToken1”, ConfigurationToken = “VECToken1”, where “VECToken1” is video encoder configuration from **GetCompatibleVideoEncoderConfigurationsResponse** message which is not present in already selected profiles) to add video encoder configuration to profile. Use this profile as result of procedure.

17. Repeat steps 4-17 to create required number of profiles.

Note: See Annex A.22 for Name and Token Parameters Length limitations.

A.15 Start Media Stream Using RTP-Unicast/UDP Transport

Name: HelperStreamingRTPUnicastUDP

Procedure Purpose: Helper procedure to verify media streaming over RTP-Unicast/UDP.

Pre-requisite: None

Input: Uri for media streaming (*streamUri*) OR Media profile token (*profileToken*). Media type (*mediaType*). Expected media stream encoding (*encoding*). Media type2 (*mediaType2*) (optional parameter). Expected media stream encoding for Media type2 (*encoding2*) (optional parameter).

Returns: None

Procedure:

1. If *streamUri* is not provided, ONVIF Client retrieves a stream uri for Media Profile for required transport protocol by following the procedure mentioned in Annex A.21 with the following input and output parameters
   - in RTP-Unicast - Stream Type
   - in UDP - Transport protocol
   - in *profileToken* - Media profile token
   - out *streamUri* - Stream URI
2. ONVIF Client checks type of stream URI by following the procedure mentioned in Annex A.30 with the following input parameters
   - in *streamUri* - Uri for media streaming
3. ONVIF Client invokes **RTSP DESCRIBE** request to *streamUri* address.
4. The DUT responds with **200 OK** message with parameters
   - SDP information =: *sdp*
   - Header of response to DESCRIBE =: *responseHeader*
5. If *sdp* does not contain Media Type = *mediaType* with rtpmap value corresponding to *encoding* and without session attribute "sendonly" (a=sendonly), FAIL the test and skip other steps.
6. If *mediaType2* is specified and *sdp* does not contain Media Type = *mediaType2* with rtpmap value corresponding to *encoding2*, FAIL the test and skip other steps.

7. ONVIF Client checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31 with the following input parameters
   - in *responseHeader* - header of response to DESCRIBE
   - in *sdp* - SDP information
   - in *streamUri* - Uri for media streaming
8. ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType* media type (see [RFC2326] for details), with parameters
   - Transport := RTP/AVP;unicast;client_port=*port1*-*port2*
9. The DUT responds with **200 OK** message with parameters
   - Transport
   - Session =: *session*
10. If *mediaType2* is specified:
    - ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType2* media type (see [RFC2326] for details), with parameters
      - Transport := RTP/AVP;unicast;client_port=*port3*-*port4*
    - The DUT responds with **200 OK** message with parameters
      - Transport
      - Session =: *session*
11. ONVIF Client invokes **RTSP PLAY** request to uri address, which corresponds to aggregate control (see [RFC2326] for details), with parameters
    - Session := *session*
12. The DUT responds with **200 OK** message with parameters
    - Session
    - RTP-Info
13. If DUT does not send *encoding* RTP media stream to ONVIF Client over UDP, FAIL the test and skip other steps.
14. If *mediaType2* is specified and DUT does not send *encoding2* RTP media stream to ONVIF Client over UDP, FAIL the test and skip other steps.

15. If DUT does not send valid RTCP packets, FAIL the test and skip other steps.

16. ONVIF Client invokes **RTSP TEARDOWN** request to uri address, which corresponds to aggregate control (see [RFC2326] for details), with parameters

    - Session := *session*

17. The DUT responds with **200 OK** message with parameters

    - Session

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

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

- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.1 for invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: If *encoding* = AAC, then rtpmap value may be equal either MP4A-LATM or MPEG4-GENERIC at step 5 and step 6.

A.16 Start Media Stream Using RTP-Multicast/UDP Transport

Name: HelperStreamingRTPMulticastUDP

Procedure Purpose: Helper procedure to verify media streaming over RTP-Multicast/UDP.

Pre-requisite: None

Input: Uri for media streaming (*streamUri*) OR Media profile token (*profileToken*). Media type (*mediaType*). Expected media stream encoding (*encoding*). Media type2 (*mediaType2*) (optional parameter). Expected media stream encoding for Media type2 (*encoding2*) (optional parameter).

Returns: None

Procedure:

1. If required, configure multicast settings for other entities from profile by following the procedure mentioned in Annex A.11.

2. If *streamUri* is not provided, ONVIF Client retrieves a stream uri for Media Profile for required transport protocol by following the procedure mentioned in Annex A.21 with the following input and output parameters

   - in RTP-Multicast - Stream Type

   - in UDP - Transport protocol

   - in *profileToken* - Media profile token

   - out *streamUri* - Stream URI

3. ONVIF Client checks type of stream URI by following the procedure mentioned in Annex A.30 with the following input parameters

   - in *streamUri* - Uri for media streaming

4. ONVIF Client invokes **RTSP DESCRIBE** request to *streamUri* address.

5. The DUT responds with **200 OK** message with parameters

   - SDP information =: *sdp*

   - Header of response to DESCRIBE =: *responseHeader*

6. If *sdp* does not contain Media Type = *mediaType* with rtpmap value corresponding to *encoding* and without session attribute "sendonly" (a=sendonly), FAIL the test and skip other steps.

7. If *mediaType2* is specified and *sdp* does not contain Media Type = *mediaType2* with rtpmap value corresponding to *encoding2*, FAIL the test and skip other steps.

8. ONVIF Client checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31 with the following input parameters

   - in *responseHeader* - header of response to DESCRIBE

   - in *sdp* - SDP information

   - in *streamUri* - Uri for media streaming

9. ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType* media type (see [RFC2326] for details), with parameters

   - Transport := RTP/AVP;unicast;client_port=*port1*-*port2*

10. The DUT responds with **200 OK** message with parameters
    - Transport
    - Session =: *session*
11. If *mediaType2* is specified:
    - ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType2* media type (see [RFC2326] for details), with parameters
        - Transport := RTP/AVP;unicast;client_port=*port3*-*port4*
    - The DUT responds with **200 OK** message with parameters
        - Transport
        - Session =: *session*
12. ONVIF Client invokes **RTSP PLAY** request to uri address, which corresponds to aggregate control (see [RFC2326] for details), with parameters
    - Session := *session*
13. The DUT responds with **200 OK** message with parameters
    - Session
    - RTP-Info
14. If DUT does not send *encoding* RTP media stream to ONVIF Client over UDP, FAIL the test and skip other steps.
15. If *mediaType2* is specified and DUT does not send *encoding2* RTP media stream to ONVIF Client over UDP, FAIL the test and skip other steps.
16. If DUT does not send valid RTCP packets, FAIL the test and skip other steps.
17. ONVIF Client invokes **RTSP TEARDOWN** request to uri address, which corresponds to aggregate control (see [RFC2326] for details), with parameters
    - Session := *session*
18. The DUT responds with **200 OK** message with parameters
    - Session

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **RTSP 200 OK** response for **RTSP DESCRIBE**, **RTSP SETUP**, **RTSP PLAY** and **RTSP TEARDOWN** requests.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.1 for invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: If *encoding* = AAC, then rtpmap value may be equal either MP4A-LATM or MPEG4-GENERIC at step 6 and step 7.

A.17 Start Media Stream Using RTP-Unicast/RTSP/HTTP/TCP Transport

Name: HelperStreamingRTPUnicastRTSPHTTPTCP

Procedure Purpose: Helper procedure to verify media streaming over RTP-Unicast/RTSP/HTTP/TCP.

Pre-requisite: None

Input: Uri for media streaming (*streamUri*) OR Media profile token (*profileToken*). Media type (*mediaType*). Expected media stream encoding (*encoding*). Media type2 (*mediaType2*) (optional parameter). Expected media stream encoding for Media type2 (*encoding2*) (optional parameter).

Returns: None

Procedure:

1. If *streamUri* is not provided, ONVIF Client retrieves a stream uri for Media Profile for required transport protocol by following the procedure mentioned in Annex A.21 with the following input and output parameters
    - in RTP-Unicast - Stream Type
    - in HTTP - Transport protocol
    - in *profileToken* - Media profile token

- out *streamUri* - Stream URI

2. ONVIF Client checks type of stream URI by following the procedure mentioned in Annex A.30 with the following input parameters

    - in *streamUri* - Uri for media streaming

3. ONVIF Client invokes **HTTP GET** request to *streamUri* address to establish DUT to ONVIF Client connection for RTP data transfer (*connection1*).

4. ONVIF Client invokes **HTTP POST** request to *streamUri* address to establish ONVIF Client to DUT connection for RTSP control requests (*connection2*).

5. ONVIF Client invokes **RTSP DESCRIBE** request to *streamUri* address converted to rtsp address on *connection2*.

6. The DUT responds with **200 OK** message with parameters on *connection1*

    - SDP information =: *sdp*

    - Header of response to DESCRIBE =: *responseHeader*

7. If *sdp* does not contain Media Type = *mediaType* with rtpmap value corresponding to *encoding* and without session attribute "sendonly" (a=sendonly), FAIL the test and skip other steps.

8. If *mediaType2* is specified and *sdp* does not contain Media Type = *mediaType2* with rtpmap value corresponding to *encoding2* and without session attribute "sendonly" (a=sendonly), FAIL the test and skip other steps.

9. ONVIF Client checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31 with the following input parameters

    - in *responseHeader* - header of response to DESCRIBE

    - in *sdp* - SDP information

    - in *streamUri* - Uri for media streaming

10. ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType* media type (see [RFC2326] for details) on *connection2*, with parameters

    - Transport := RTP/AVP/TCP;unicast;client_port=*port1-port2*

11. The DUT responds with **200 OK** message on *connection1* with parameters

    - Transport

- Session =: *session*

12. If *mediaType2* is specified:

    - ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType2* media type (see [RFC2326] for details), with parameters

        - Transport := RTP/AVP/TCP;unicast;client_port=*port3-port4*

    - The DUT responds with **200 OK** message with parameters

        - Transport

        - Session =: *session*

13. ONVIF Client invokes **RTSP PLAY** request to uri address, which corresponds to aggregate control (see [RFC2326] for details) on *connection2*, with parameters

    - Session := *session*

14. The DUT responds with **200 OK** message on *connection1* with parameters

    - Session

    - RTP-Info

15. If DUT does not send *encoding* RTP media stream to ONVIF Client over *connection1*, FAIL the test and skip other steps.

16. If *mediaType2* is specified and DUT does not send *encoding2* RTP media stream to ONVIF Client over *connection1*, FAIL the test and skip other steps.

17. If DUT does not send valid RTCP packets, FAIL the test and skip other steps.

18. ONVIF Client invokes **RTSP TEARDOWN** request to uri address, which corresponds to aggregate control (see [RFC2326] for details) on *connection2*, with parameters

    - Session := *session*

19. ONVIF Client closes *connection2*.

20. The DUT responds with **HTTP 200 OK** message on *connection1* and closes *connection1*.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **RTSP 200 OK** response for **RTSP DESCRIBE**, **RTSP SETUP** and **RTSP PLAY** requests.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.1 for invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: If *encoding* = AAC, then rtpmap value may be equal either MP4A-LATM or MPEG4-GENERIC at step 7 and step 8.

A.18 Start Media Stream Using RTP/RTSP/TCP Transport

Name: HelperStreamingRTPRTSPTCP

Procedure Purpose: Helper procedure to verify media streaming over RTP/RTSP/TCP.

Pre-requisite: None

Input: Uri for media streaming (*streamUri*) OR Media profile token (*profileToken*). Media type (*mediaType*). Expected media stream encoding (*encoding*). Media type2 (*mediaType2*) (optional parameter). Expected media stream encoding for Media type2 (*encoding2*) (optional parameter).

Returns: None

Procedure:

1. If *streamUri* is not provided, ONVIF Client retrieves a stream uri for Media Profile for required transport protocol by following the procedure mentioned in Annex A.21 with the following input and output parameters
   - in RTP-Unicast - Stream Type
   - in RTSP - Transport protocol
   - in *profileToken* - Media profile token
   - out *streamUri* - Stream URI
2. ONVIF Client checks type of stream URI by following the procedure mentioned in Annex A.30 with the following input parameters
   - in *streamUri* - Uri for media streaming

3. ONVIF Client invokes **RTSP DESCRIBE** request to *streamUri* address.

4. The DUT responds with **200 OK** message with parameters

   - SDP information =: *sdp*

   - Header of response to DESCRIBE =: *responseHeader*

5. If *sdp* does not contain Media Type = *mediaType* with rtpmap value corresponding to *encoding* and without session attribute "sendonly" (a=sendonly), FAIL the test and skip other steps.

6. If *mediaType2* is specified and *sdp* does not contain Media Type = *mediaType2* with rtpmap value corresponding to *encoding2* and without session attribute "sendonly" (a=sendonly), FAIL the test and skip other steps.

7. ONVIF Client checks types of IP addresses returned in response to DESCRIBE by following the procedure mentioned in Annex A.31 with the following input parameters

   - in *responseHeader* - header of response to DESCRIBE

   - in *sdp* - SDP information

   - in *streamUri* - Uri for media streaming

8. ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType* media type (see [RFC2326] for details), with parameters

   - Transport := RTP/AVP/TCP;unicast;interleaved=0-1

9. The DUT responds with **200 OK** message with parameters

   - Transport

   - Session =: *session*

10. If *mediaType2* is specified:

    - ONVIF Client invokes **RTSP SETUP** request to uri address, which corresponds to *mediaType2* media type (see [RFC2326] for details), with parameters

      - Transport := RTP/AVP/TCP;unicast;interleaved=0-1

    - The DUT responds with **200 OK** message with parameters

      - Transport

      - Session =: *session*

11. ONVIF Client invokes **RTSP PLAY** request to uri address, which corresponds to aggregate control (see [RFC2326] for details), with parameters
    - Session := *session*
12. The DUT responds with **200 OK** message with parameters
    - Session
    - RTP-Info
13. If DUT does not send *encoding* RTP media stream to ONVIF Client over RTSP control connection, FAIL the test and skip other steps.
14. If *mediaType2* is specified and DUT does not send *encoding2* RTP media stream to ONVIF Client over RTSP control connection, FAIL the test and skip other steps.
15. If DUT does not send valid RTCP packets, FAIL the test and skip other steps.
16. ONVIF Client invokes **RTSP TEARDOWN** request to uri address, which corresponds to aggregate control (see [RFC2326] for details), with parameters
    - Session := *session*
17. The DUT responds with **200 OK** message with parameters
    - Session

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **RTSP 200 OK** response for **RTSP DESCRIBE**, **RTSP SETUP**, **RTSP PLAY** and **RTSP TEARDOWN** requests.
- RTSP Session is terminated by DUT during media streaming.

Note: See Annex A.1 for invalid RTP header definition.

Note: ONVIF Client checks authentication type for all RTSP requests by following the procedure mentioned in Annex A.26.

Note: If *encoding* = AAC, then rtpmap value may be equal either MP4A-LATM or MPEG4-GENERIC at step 5 and step 6.

A.19 Media Profile Select for JPEG Video Streaming

For the execution of real time streaming - video test cases, ONVIF Client has to select the media profile with Video Source Configuration and Video Encoder Configuration based on the JPEG video codec.

ONVIF Client follows the following procedure to select the media profile:

1. Retrieve media profiles by invoking GetProfiles command.
2. Select media profiles that contain Video Encoder Configuration and Video Source Configuration.
3. Retrieve supported video encoder configuration options for a media profile by invoking GetVideoEncoderConfigurationOptions (media profile token) command. Check whether the selected media profile supports the JPEG video codec.
4. Repeat step 3 for all media profiles selected on step 2 till a media profile with the JPEG video codec support is found.

A.20 Media Profile Configuration with Required Audio Encoder and with Video Encoder

Name: HelperProfileConfigurationWithAudioAndVideo

Procedure Purpose: Helper procedure to configure Media Profile with Audio Source Configuration, Audio Encoder Configuration with required encoder, Video Source Configuration and Video Encoder Configuration.

Pre-requisite: Media Service is received from the DUT. Audio is supported by the DUT.

Input: Required audio encoding (*requiredAudioEncoding*).

Returns: Media profile with required configurations (*profile*).

Procedure:

1. ONVIF Client invokes **GetProfiles** request
2. DUT responds with **GetProfilesResponse** message with parameters
   - Profiles list =: *profileList*
3. If *profileList* contains Profile item (*profile1*) with VideoSourceConfiguration, VideoEncoderConfiguration, AudioSourceConfiguration, and AudioEncoderConfiguration:
   - If *profile1*.AudioEncoderConfiguration.Encoding = *requiredAudioEncoding*:

- set *profile* := *profile1* and skip other steps.

4. ONVIF Client configures an empty media profile by following the procedure mentioned in Annex A.9 with the following input and output parameters

   - out *profile* - empty media profile

5. ONVIF Client invokes **GetCompatibleAudioSourceConfigurations** request with parameters

   - ProfileToken := *profile*.@token

6. DUT responds with **GetCompatibleAudioSourceConfigurationsResponse** message with parameters

   - Configurations list =: *ascList*

7. If *ascList* is empty, restore DUT settings, FAIL the test and skip other steps.

8. For each Audio Source Configuration (*asc*) in *ascList* repeat the following steps:

   8.1. ONVIF Client invokes **AddAudioSourceConfiguration** request with parameters

   - ProfileToken := *profile*.@token

   - ConfigurationToken := *asc*.@token

   8.2. DUT responds with **AddAudioSourceConfigurationResponse** message

   8.3. ONVIF Client invokes **GetCompatibleAudioEncoderConfigurations** request with parameters

   - ProfileToken := *profile*.@token

   8.4. DUT responds with **GetCompatibleAudioEncoderConfigurationsResponse** message with parameters

   - Configurations list =: *aecList*

   8.5. For each Audio Encoder Configuration (*aec*) in *aecList* repeat the following steps:

   8.5.1. ONVIF Client invokes **GetAudioEncoderConfigurationOptions** request with parameters

   - ConfigurationToken := *aec*.@token

   - ProfileToken := *profile*.@token

8.5.2. DUT responds with **GetAudioEncoderConfigurationOptionsResponse** message with parameters

- Options list =: *optionsList*

8.5.3. If *optionsList* list contains item with Encoding = *requiredAudioEncoding*:

8.5.3.1. ONVIF Client invokes **AddAudioEncoderConfiguration** request with parameters

- ProfileToken := *profile*.@token
- ConfigurationToken := *aec*.@token

8.5.3.2. The DUT responds with **AddAudioEncoderConfigurationResponse** message.

8.5.3.3. Set *aecOptions* := item with Encoding = *requiredAudioEncoding* from *optionsList* list.

8.5.3.4. ONVIF Client invokes **SetAudioEncoderConfiguration** request with parameters

- Configuration.@token := *aec*.@token
- Configuration.Name := *aec*.Name
- Configuration.UseCount := *aec*.UseCount
- Configuration.Encoding := *requiredAudioEncoding*
- Configuration.Bitrate := the nearest value to *aec*.Bitrate from *aecOptions*BitrateList.Items list
- Configuration.SampleRate := the nearest value to *aec*.SampleRate from *aecOptions*SampleRateList.Items list
- Configuration.Multicast := *aec*.Multicast
- Configuration.SessionTimeout := *aec*.SessionTimeout
- ForcePersistence := false

8.5.3.5. The DUT responds with **SetAudioEncoderConfigurationResponse** message.

8.5.3.6. ONVIF Client invokes **GetCompatibleVideoSourceConfigurations** request with parameters

- ProfileToken := *profile*.@token

8.5.3.7. DUT responds with **GetCompatibleVideoSourceConfigurationsResponse** message with parameters

- Configurations list =: *vscList*

8.5.3.8. If *vscList* is empty:

- ONVIF Client invokes **RemoveAudioEncoderConfiguration** request with parameters
  - ProfileToken := *profile*.@token
- DUT responds with **RemoveAudioEncoderConfigurationResponse** message
- Go to step 8.

8.5.3.9. For each Video Source Configuration *vsc* in *vscList* repeat the following steps:

8.5.3.9.1. ONVIF Client invokes **AddVideoSourceConfiguration** request with parameters

- ProfileToken := *profile*.@token
- ConfigurationToken := *vsc*.@token

8.5.3.9.2. The DUT responds with **AddVideoSourceConfigurationResponse** message.

8.5.3.9.3. ONVIF Client invokes **GetCompatibleVideoEncoderConfigurations** request with parameters

- ProfileToken := *profile*.@token

8.5.3.9.4. DUT responds with **GetCompatibleVideoEncoderConfigurationsResponse** message with parameters

- Configurations list =: *vecList*

8.5.3.9.5. If *vecList* is empty and *vsc* is the last item in *vscList*:

- ONVIF Client invokes **RemoveVideoSourceConfiguration** request with parameters
  - ProfileToken := *profile*.@token
- DUT responds with **RemoveVideoSourceConfigurationResponse** message
- ONVIF Client invokes **RemoveAudioEncoderConfiguration** request with parameters
  - ProfileToken := *profile*.@token
- DUT responds with **RemoveAudioEncoderConfigurationResponse** message
- Go to step 8.

8.5.3.9.6. If *vecList* is not empty

- ONVIF Client invokes **AddVideoEncoderConfiguration** request with parameters
  - ProfileToken := *profile*.@token

- ConfigurationToken := *vecList*[0].@token

- DUT responds with **AddVideoEncoderConfigurationResponse** message

- Skip other steps.

9. If profile with Audio Source Configuration, Audio Encoder Configuration with required encoder, Video Source Configuration and Video Encoder Configuration was not configured at step 8, restore DUT settings, FAIL the test and skip other steps.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetProfilesResponse** message.
- DUT did not send **GetCompatibleAudioSourceConfigurationsResponse** message.
- DUT did not send **AddAudioSourceConfigurationResponse** message.
- DUT did not send **GetCompatibleAudioEncoderConfigurationsResponse** message.
- DUT did not send **GetAudioEncoderConfigurationOptionsResponse** message.
- DUT did not send **AddAudioEncoderConfigurationResponse** message.
- DUT did not send **SetAudioEncoderConfigurationResponse** message.
- DUT did not send **GetCompatibleVideoSourceConfigurationsResponse** message.
- DUT did not send **RemoveAudioEncoderConfigurationResponse** message.
- DUT did not send **AddVideoSourceConfigurationResponse** message.
- DUT did not send **GetCompatibleVideoEncoderConfigurationsResponse** message.
- DUT did not send **AddVideoEncoderConfigurationResponse** message.
- DUT did not send **RemoveVideoSourceConfigurationResponse** message.

A.21 Get Stream Uri

Name: HelperGetStreamUri

Procedure Purpose: Helper procedure to get stream URI from the DUT.

Pre-requisite: Media Service is received from the DUT.

Input: Stream Type (*streamType*), Protocol (*protocol*), Media Profile token (*token*)

Returns: Stream Uri (*streamUri*).

Procedure:

1. ONVIF Client invokes **GetStreamUri** request with parameters
   - StreamSetup.Stream := *streamType*
   - StreamSetup.Transport.Protocol := *protocol*
   - ProfileToken := *token*
2. The DUT responds with **GetStreamUriResponse** message with parameters
   - MediaUri.Uri =: *streamUri*
   - MediaUri.InvalidAfterConnect
   - MediaUri.InvalidAfterReboot
   - MediaUri.Timeout
3. If *streamUri* is longer than 128 octets, FAIL the test and skip other steps.
4. If *protocol* = HTTP:

   4.1. If *streamUri* doesn't have the same scheme with the web service ('http'), FAIL the test and skip other steps.
5. If *protocol* != HTTP:

   5.1. If *streamUri* doesn't have scheme equal to 'rtsp', FAIL the test and skip other steps.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetStreamUriResponse** message.

A.22 Name and Token Parameters

There are the following limitations on maximum length of the Name and Token parameters that shall be used during tests by ONVIF Device Test Tool to prevent faults from DUT:

- Name shall be less than or equal to 64 characters (only readable characters accepted).
- Token shall be less than or equal to 64 characters (only readable characters accepted).
- UTF-8 character set shall be used for Name and Token.

Note: these limitations will not be used, if ONVIF Device Test Tool reuses values that were received from the DUT.

A.23 Turn on IPv6 network interface

Name: HelperTurnOnIPv6

Procedure Purpose: Helper procedure to turn on IPv6 network interface.

Pre-requisite: IPv6 is supported by DUT.

Input: None

Returns: Initial Network settings (*initialNetworkSettings*).

Procedure:

1. ONVIF Client will invoke **GetNetworkInterfacesRequest** message to retrieve the original settings of the DUT.
2. ONVIF Client verifies **GetNetworkInterfacesResponse** message.
3. Set *initialNetworkSettings* := available network interface.
4. If **GetNetworkInterfacesResponse** message contains NetworkInterfaces.IPv6 and NetworkInterfaces.IPv6.Enabled=true, then ONVIF Client checks NetworkInterfaces.IPv6.Config.DHCP. Otherwise, go to step 11.
5. If NetworkInterfaces.IPv6.Config.DHCP=Off, then ONVIF Client checks NetworkInterfaces.IPv6.Config.Manual element. Otherwise, go to step 8.
6. If NetworkInterfaces.IPv6.Config.Manual element is present and not empty, then ONVIF Client skips other steps and run test using NetworkInterfaces.IPv6.Config.Manual value as device IP. Otherwise, ONVIF Client checks NetworkInterfaces.IPv6.Config.LinkLocal element.

7. If NetworkInterfaces.IPv6.Config.LinkLocal element is present and not empty, then ONVIF Client skips other steps and runs test using NetworkInterfaces.IPv6.Config.LinkLocal value as device IP. Otherwise, ONVIF Client skip other steps and failed test.

8. ONVIF Client will invoke **SetNetworkInterfacesRequest** message to turn off DHCP IPv6 (InterfaceToken = available network interface, NetworkInterfaces.IPv6.Config.DHCP=Off).

9. ONVIF Client gets current network interfaces via **GetNetworkInterfacesRequest** message.

10. ONVIF Client verifies **GetNetworkInterfacesResponse** message and checks that set settings were applied. Repeat steps 6-7.

11. If **GetNetworkInterfacesResponse** message does not contain NetworkInterfaces.IPv6 or NetworkInterfaces.IPv6.Enabled=false, then ONVIF Client will invoke SetNetworkInterfacesRequest message (InterfaceToken = available network interface, NetworkInterfaces.IPv6. Enabled=true) to turn on IPv6 configuration.

12. The DUT will return **SetNetworkInterfacesResponse** message.

13. If Reboot is required by DUT, invoke **SystemReboot** command.

14. If DUT supports Discovery:

    14.1. ONVIF Client waits for HELLO message from the default network interface.

15. If DUT does not support Discovery:

    15.1. ONVIF Client waits during *rebootTimeout*.

16. ONVIF Client gets current network interfaces via **GetNetworkInterfacesRequest** message.

17. ONVIF Client verifies **GetNetworkInterfacesResponse** message and checks that set settings were applaied. Execute steps 5-7.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

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

- DUT did not send **SetNetworkInterfacesResponse** message.

- DUT did not send **SystemReboot** message.

A.24 Restore Network Settings

Name: HelperRestoreNetworkSettings

Procedure Purpose: Helper procedure to restore the original default settings.

Pre-requisite: None

Input: Initial Network settings to restore (*initialNetworkSettings*).

Returns: None

Procedure:

1. Restore the initial network settings by invoking **SetNetworkInterfaces** (Default settings) command.

2. If Reboot is required by DUT, invoke **SystemReboot** command.

3. If **SystemReboot** is invoked and DUT supports Discovery, wait for HELLO message from the default network interface.

4. If DUT does not support Discovery:

   4.1. ONVIF Client waits during *rebootTimeout*.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

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

- DUT did not send **SetNetworkInterfacesResponse** message.

- DUT did not send **SystemReboot** message.

A.25 Media Profile Configuration for Backchannel Audio Streaming

For the execution of real time streaming – Audio Backchannel test cases, ONVIF Client has to select and configure the media profile based on the required audio decoder.

ONVIF Client follows the procedure to configure the media profile:

1. Retrieve media profiles by invoking **GetProfiles** command.

2. If media profile includes audio output and audio decoder configurations:

   a. Retrieve the supported audio decoder configuration options for the media profile by invoking **GetAudioDecoderConfigurationOptions** (media profile token) command. Check whether the selected media profile supports the required audio decoder.

   b. Retrieve the supported audio output configuration options for the media profile by invoking **GetAudioOutputConfigurationOptions** (media profile token) command. Check whether the selected media profile supports the required send primacy.

3. If the media profile does not have audio source and audio encoder configuration:

   a. Retrieve audio output configurations of the DUT by invoking **GetAudioOutputConfigurations** command.

   b. Retrieve audio output configuration options supported for an audio output configuration by invoking **GetAudioOutputConfigurationOptions** (audio output configuration token) command. Check whether the selected audio output configuration supports the required send primacy.

   c. Add audio output configuration to the profile by invoking **AddAudioOutputConfigurations** command.

   d. Repeat test procedure – 3.b for all audio output configurations till an audio output configuration with the required send primacy is found.

   e. Retrieve audio decoder configurations of the DUT by invoking **GetAudioDecoderConfigurations** command.

   f. Retrieve audio decoder configuration options supported for an audio decoder configuration by invoking **GetAudioDecoderConfigurationOptions** (audio decoder configuration token) command. Check whether the selected audio decoder configuration supports the required audio decoder.

   g. Repeat test procedure – 3.f for all audio decoder configurations till an audio decoder configuration with the required audio decoder is found.

   h. Add the corresponding audio decoder configuration to the media profile by invoking **AddAudioDecoderConfiguration** command.

4. Repeat test procedure – 2 and 3 for all media profiles till a media profile with the required audio output and audio decoder support is found.

A.26 RTSP Authentication Check

Name: HelperRTSPAuthenticationCheck

Procedure Purpose: Helper procedure to check that DUT uses Digest authentication type for RTSP.

Pre-requisite: Real-time Streaming is supported by DUT.

Input: None.

Returns: None.

Procedure:

1. If Device supports Profile T according to feature definition results:

   1.1. If DUT does not send **RTSP 401 Unauthorized** message to any RTSP anonymous request, FAIL the test and skip other steps.

   1.2. If **RTSP 401 Unauthorized** message does not have **WWW-Authenticate: Digest** header, FAIL the test and skip other steps.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send response message to RTSP request.

A.27 Video Encoder Configuration – Resolution Selection

For all video test cases, ONVIF Client selects resolution nearest to 640x480 from video encoder configuration options and set it during configuration procedure, with the following exceptions:

- RTSS-1-1-24 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION (ALL RESOLUTIONS)
- RTSS-1-1-25 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION (ALL RESOLUTIONS)
- RTSS-1-1-26 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION (ALL RESOLUTIONS)

- RTSS-1-1-46 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION
- RTSS-1-1-47 VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION
- RTSS-1-1-48 VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION
- RTSS-1-1-53 MEDIA STREAMING – JPEG (VALIDATING RTP HEADER EXTENSION)

A.28 Video Encoder Configuration – Frame Rate Selection

For all video test cases, ONVIF Client selects framerate nearest to 25 from video encoder configuration options and set it during configuration procedure, with the following exceptions:

- RTSS-1-1-24 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION (ALL RESOLUTIONS)
- RTSS-1-1-46 VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION
- RTSS-1-1-53 MEDIA STREAMING – JPEG (VALIDATING RTP HEADER EXTENSION)

For listed test cases, ONVIF Client selects framerate nearest to 5 from video encoder configuration options and set it during configuration procedure for highest JPEG resolution.

A.29 Removing Configurations from Media Profile

Name: HelperRemoveConfigurationsFromMediaProfile

Procedure Purpose: Helper Procedure to remove configuartions from Media Profile.

Pre-requisite: Media Service is received from the DUT.

Input: Media Profile (*profile*). List of configuration type to remove from profile (*confTypeList*).

Returns: None.

Procedure:

1. If *confTypeList* contains VideoEncoderConfiguration and *profile* contains VideoEncoderConfiguration:

   1.1. ONVIF Client invokes **RemoveVideoEncoderConfiguration** request with parameters

   - ProfileToken := *profile*.@token

   1.2. The DUT responds with **RemoveVideoEncoderConfigurationResponse** message.

2. If *confTypeList* contains AudioEncoderConfiguration and *profile* contains AudioEncoderConfiguration:

2.1. ONVIF Client invokes **RemoveAudioEncoderConfiguration** request with parameters

- ProfileToken := *profile*.@token

2.2. The DUT responds with **RemoveAudioEncoderConfigurationResponse** message.

3. If *confTypeList* contains MetadataConfiguration and *profile* contains MetadataConfiguration:

3.1. ONVIF Client invokes **RemoveMetadataConfiguration** request with parameters

- ProfileToken := *profile*.@token

3.2. The DUT responds with **RemoveMetadataConfigurationResponse** message.

4. If *confTypeList* contains VideoSourceConfiguration and *profile* contains VideoSourceConfiguration:

4.1. ONVIF Client invokes **RemoveVideoSourceConfiguration** request with parameters

- ProfileToken := *profile*.@token

4.2. The DUT responds with **RemoveVideoSourceConfigurationResponse** message.

5. If *confTypeList* contains AudioSourceConfiguration and *profile* contains AudioSourceConfiguration:

5.1. ONVIF Client invokes **RemoveAudioSourceConfiguration** request with parameters

- ProfileToken := *profile*.@token

5.2. The DUT responds with **RemoveAudioSourceConfigurationResponse** message.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **RemoveVideoEncoderConfigurationResponse** message.
- DUT did not send **RemoveAudioEncoderConfigurationResponse** message.
- DUT did not send **RemoveMetadataConfigurationResponse** message.
- DUT did not send **RemoveAudioSourceConfigurationResponse** message.

- DUT did not send **RemoveVideoSourceConfigurationResponse** message.

A.30 Check IP Type of URI

Name: HelperCheckIPTypeOfURI

Procedure Purpose: Helper procedure to check type of URI.

Pre-requisite: None.

Input: URI (*uri*), IP type *ipType*.

Returns: None.

Procedure:

1. If type of host part of *uri* is not equal to *ipType*, FAIL the test and skip other steps.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- None.

A.31 Check of IP address type in response to RTSP DESCRIBE

Name: HelperIPAddressTypeInRTSP

Procedure Purpose: Helper procedure to check IP addresses types returned by DUT in response to RTSP DESCRIBE.

Pre-requisite: None.

Input: Header of response to DESCRIBE (*responseHeader*). SDP (*sdp*). Stream Uri (*streamUri*).

Returns: None.

Procedure:

1. Set *ipType* := *streamUri* IP type.
2. For each **Content-Base** field in *responseHeader* (*contentBase*) that has absolute IP value:

   2.1. If *contentBase* IP value does not correspond to *ipType*, FAIL the test and skip other steps (see [RFC2326] for details).

3. For each **Content-Location** field in *responseHeader* (*contentLocation*) that has absolute IP value:

   3.1. If *contentLocation* IP value does not correspond to *ipType*, FAIL the test and skip other steps (see [RFC2326] for details).

4. For each **"a=control"** attribute in *sdp* (*aControl*) that has absolute IP value:

   4.1. If *aControl* IP value does not correspond to *ipType*, FAIL the test and skip other steps (see [RFC2326] for details).

5. If *ipType* = IPv4:

   5.1. If *sdp* contains at least one origin field ("o=") with **addrtype** != "IP4", FAIL the test and skip other steps (see [RFC4566] for details).

   5.2. If *sdp* contains at least one origin field ("o=") with IP type of **unicast-address** sub-field != IPv4 type, FAIL the test and skip other steps (see [RFC4566] for details).

   5.3. If *sdp* contains at least one connection data field ("c=") with **addrtype** != "IP4", FAIL the test and skip other steps (see [RFC4566] for details).

   5.4. If *sdp* contains at least one connection data field ("c=") with IP type of **connection address** sub-field != IPv4 type, FAIL the test and skip other steps (see [RFC4566] for details).

6. If *ipType* = IPv6:

   6.1. If *sdp* contains at least one origin field ("o=") with **addrtype** != "IP6", FAIL the test and skip other steps (see [RFC4566] for details).

   6.2. If *sdp* contains at least one origin field ("o=") with IP type of **unicast-address** sub-field != IPv6 type, FAIL the test and skip other steps (see [RFC4566] for details).

   6.3. If *sdp* contains at least one connection data field ("c=") with **addrtype** != "IP6", FAIL the test and skip other steps (see [RFC4566] for details).

   6.4. If *sdp* contains at least one connection data field ("c=") with IP type of **connection address** sub-field != IPv6 type, FAIL the test and skip other steps (see [RFC4566] for details).

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- None.