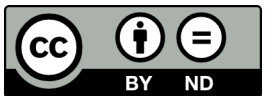
Video Services Forum (VSF)
Test Plan TR-10 TP-1

Internet Protocol Media Experience (IPMX):
IPMX Test Plan 1

January 13, 2026



INTELLECTUAL PROPERTY RIGHTS

RECIPIENTS OF THIS DOCUMENT ARE REQUESTED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT CLAIMS OR OTHER INTELLECTUAL PROPERTY RIGHTS OF WHICH THEY MAY BE AWARE THAT MIGHT BE INFRINGED BY ANY IMPLEMENTATION OF THE RECOMMENDATION SET FORTH IN THIS DOCUMENT, AND TO PROVIDE SUPPORTING DOCUMENTATION.

THIS RECOMMENDATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS RECOMMENDATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY MPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS RECOMMENDATION.

LIMITATION OF LIABILITY

VSF SHALL NOT BE LIABLE FOR ANY AND ALL DAMAGES, DIRECT OR INDIRECT, ARISING FROM OR RELATING TO ANY USE OF THE CONTENTS CONTAINED HEREIN, INCLUDING WITHOUT LIMITATION ANY AND ALL INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS, LOSS OF PROFITS, LITIGATION, OR THE LIKE), WHETHER BASED UPON BREACH OF CONTRACT, BREACH OF WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING NEGATION OF DAMAGES IS A FUNDAMENTAL ELEMENT OF THE USE OF THE CONTENTS HEREOF, AND THESE CONTENTS WOULD NOT BE PUBLISHED BY VSF WITHOUT SUCH LIMITATIONS.

Executive Summary

Internet Protocol Media Experience (IPMX) was created to foster the adoption of open standards-based protocols for interoperability over IP in the media and entertainment and professional audio/video industries. IPMX is based on the SMPTE ST 2110 standard and as such the VSF TR-10 suite of Technical Recommendations is a set of differences between SMPTE ST 2110 and IPMX.

This Technical Recommendation documents the minimum requirements for IPMX device with respect to system environment and device behavior. Some of the subject covered in this document include requirements related to network support and how IPMX devices are expected to behave when sending or receiving IPMX streams when PTP is present or when it is not.

Change Log

Date	Description of Changes
2 Dec 2025	Initial release
16 Dec 2025	Added a change log. Added note to Section 10 about NMOS test not covering some of the new requirements in the AIMS profile document.
06 Jan 2026	Removed the LLDP section as this requirement is not in any TR-10 specification Added list of contributors

Table of Contents

1 Introduction (Informative)

IPMX, which stands for IP Media Experience, is based on two families of specifications. The SMPTE ST 2110 Professional Media Over Managed IP Networks suite of standards for the transport of video, audio, and ancillary/control signals over IP networks, and the NMOS REST APIs from AMWA, which provide discovery, connection management, and control.

IPMX is an accessible, open standard that meets the needs of professional and consumer video and audio users in a wide variety of contexts while giving manufacturers and developers what they need to build low-latency, interoperable, IP based audiovisual products or applications.

This document covers the test plan for the IPMX tested event. To test compliance to IPMX's technical requirements (TR-10).

2 Contributors

The following individuals participated in the Video Services Forum IPMX working group that developed this technical recommendation.

Alain Bouchard (Matrox)	Hung Lam (EvertzAV)	Kyle Wilken (Cobalt Digital)	Ryan Wallenberg (Cobalt)
Amanda Douglass (Macnica)	Jack Douglass (PacketStorm)	Leigh Whitcomb (Meinberg)	Steve Kolta (Meinberg)
Andre Testa (Matrox)	Jason Fritz (plexusAV)	Nick Nicas (AT&T)	Thomas True (NVIDIA)
Andreas Hildebrand (RAVENNA)	Jean Lapierre (Matrox)	Palo Francisco (EvertzAV)	Tim Bruylants (IntoPix)
Andrew Starks (Macnica)	Jean-Jacques Ostiguy (Matrox)	Phil Bernal (PB Technologies Group)	Wes Simpson (LearnIPvideo.com)
Dan Holland (DHC)	Jed Deame (Nextera)	Rachit Sharma (Macnica)	
Daniel Bouquet (ANALOG WAY)	Jeroen Steendam (DekTec)	Raymond Hermans (Adeas)	
David Konz (plexusAV)	Jie Gao (Novastar)	Richard Seegull (Riedel)	

3 About the Video Services Forum

The Video Services Forum, Inc. (www.videoservicesforum.org) is an international association dedicated to video transport technologies, interoperability, quality metrics and education. The VSF is composed of service providers, users and manufacturers. The organization's activities include:

- providing forums to identify issues involving the development, engineering, installation, testing and maintenance of audio and video services;
- exchanging non-proprietary information to promote the development of video transport service technology and to foster resolution of issues common to the video services industry;
- identification of video services applications and educational services utilizing video transport services;
- promoting interoperability and encouraging technical standards for national and international standards bodies.

The VSF is an association incorporated under the Not For Profit Corporation Law of the State of New York. Membership is open to businesses, public sector organizations and individuals worldwide. For more information on the Video Services Forum or this document, please call +1 929-279-1995 or e-mail opsmgr@videoservicesforum.org.

4 Conformance Notation

Normative text describes elements of the design that are indispensable or contain the conformance language keywords: "shall," "should," or "may."

Informative text is potentially helpful to the user but not indispensable and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except the Introduction and any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed to conform to the document and from which no deviation is permitted.

The keywords "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

5 References

- TR-10-1:2024 Internet Protocol Media Experience (IPMX): System Timing and Definitions
- TR-10-2:2024 Internet Protocol Media Experience (IPMX): Uncompressed Active Video
- TR-10-3:2024 Internet Protocol Media Experience (IPMX): PCM Digital Audio
- TR-10-8:2025 Internet Protocol Media Experience (IPMX): NMOS Requirements
- TR-10-9:2025 Internet Protocol Media Experience (IPMX): Requirements for System Environment and Device Behavior
- TR-10-11:2024 Internet Protocol Media Experience (IPMX): Constant Bit-Rate Compressed Video

6 Definitions

For the purposes of this document, the terms, and definitions of VSF TR-10-1, VSF TR-10-2, VSF TR-10-3, VSF TR-10-8, VSF TR-10-9, and VSF TR-10-11

DuT	Device under test.
Disruption	Artefact on a signal that includes but is not limited to: ○ Black lines in the video essence ○ Frozen/dropped frames in the video essence ○ Mutes or pops in the audio essence ○ Discontinuity of RTP sequence numbers ○ Change in RTP timestamps by more than a duration of a video frame/audio frame.

7 General Provision

These tests are based on the following AIMS document(s)

- IPMX Product Qualification and Certification Requirements

Note: This version of the test plan does not completely cover some of the latest requirements found in the AIMS IPMX Profile Requirement documents. For example, the section titled NMOS Representation (Normative) is not completely covered by the test plan at this time. Future version will and the implementor of IPMX

should make sure their implementation fulfills the requirements of the AIMS IPMX profile documents.

8 Infrastructure Required for Testing

The following equipment is required to execute this test plan

- Network capable of supporting devices under test
- Network with multicast support IGMP v2 and V3 capable
- Network with at least two separate subnet with proper IPv4 routing in between
- DHCP server that provides IPv4 address, mask, default gateway, DNS server and search domain
- Two NMOS IS-04 Registries with mDNS support and a mean to turned off mDNS.
- NMOS IS-05 Controller software
- DNS Server with proper DNS-SD entries for 2x NMOS Registries with different priority
- PTP Grandmaster
- Reference Senders
- Reference Receivers
- Device to capture PCAP files
- Device to measure the RTCP sender report and first RTP packet jitter
- Device to measure the CMAX and VRX for IPMX.
- Wireshark (with IPMX dissectors)
    - GitHub - rpkh/ipmx-rtcp-info-dissector: Lua Wireshark post-dissector for extracting IPMX info blocks from RTCP sender reports
    - https://github.com/NEOAdvancedTechnology/smpte2110-20-dissector
    - https://www.intopix.com/blogs/post/Deep-dive-into-SMPTE-ST2110-22-with-Wireshark-Dissector
- Optional
    - PTP Track Hound
        - https://www.ptptrackhound.com/#/home
    - A software to validate an SDP file content such as SD Poker
        - GitHub - AMWA-TV/sdpoker: A patched version of Streampunk/sdpoker with additional source-filter testing and bug fixes, used by AMWA-TV/nmos-testing

9 PTP configuration used

Some of the tests are executed while there is no PTP Grandmaster on the network.

Some of the tests are executed while there is a PTP Grandmaster on the network and the network configured without support for Boundary or Transparent Clock mode.
Multicast communication mode for all messages will be used (except for management TLV responses). The PTP profile details vary and are specified in more details in the relevant sections.

10 General statements and terms

- This test plan outlines the principles and methods for the IPMX senders and receivers to be tested for compliance with AIMS IPMX Profiles.
- Standard MTU size (1500 octets) will always be expected.
- Throughout the text the term “Disruption(s)” is used, it may be applied to baseband signals and/or IP streams.

11 General Network Interface Tests

Description: This set of Tests is expected to validate the general network-related functionality of a media device.

At the time of testing, Testers will provide the Vendor representative with a set of parameters consisting of:

- A unicast host IPv4 address
- A subnet mask from the range from /8 to /30
- A default gateway IPv4 address

Sets of parameters will be provided for essence interface(s) and for management interface(s) (if an out-of-band management interface is present for the DuT). The Vendor representative will be expected to configure the DuT with the given sets of parameters. Additionally, the DHCP server will be active in the network and it will be required for the DuT to demonstrate support for DHCP server. When testing an interface, the network administrator may disconnect other DuT interfaces from the network to avoid routing issues caused by multiple interfaces having default gateways.

11.1 Manual Configuration of Management Network Interface Test*

*Only applied to devices with out-of-band management interface(s).

Description: Tests the ability of DuT to configure the IPv4 address, subnet mask, and default gateway parameters of the management interface manually. Also verifies that ICMP messages are not blocked and the TTL value of ICMP replies. The DuT will be pinged from a host in another subnet. DuT is expected to reply to ICMP echo requests (ping) coming from another subnet, while properly utilizing a default gateway, and with TTL > 16. This test will not involve an IP address change, a demonstration of basic configurability will suffice.

Validation method: Console log from a device sending ICMP echo requests is used for validation.
Pass criteria:

- The DuT assumes the parameters configured manually.
- The DuT can be pinged from a host on another subnet.
- The TTL value of ICMP echo reply is > 16 for both payload sizes.

No pass criteria:

- The DuT does not assume the parameters configured manually.
- The DuT cannot be pinged from a host on another subnet with one or both payload sizes.
- The DuT does not reply with a TTL value that is > 16.

11.2 Manual Configuration of IPMX Media Network Interface Test

Description: Tests the ability of DuT to configure the IPv4 address, subnet mask, and default gateway parameters for IPMX Media network interface manually. Also verifies that ICMP messages are not blocked and the TTL value of ICMP replies. The DuT will be pinged from a host in another subnet. DuT is expected to reply to ICMP echo requests (ping) coming from another subnet, while properly utilizing a default gateway, and with TTL > 16. This test will not involve an IP address change, a demonstration of basic configurability will suffice.

Validation method: Console log from a device sending ICMP echo requests is used for validation.
Pass criteria:

- The DuT assumes the parameters configured manually.
- The DuT can be pinged from a host on another subnet.
- The TTL value of ICMP echo reply is > 16 for both payload sizes.

No pass criteria:

- The DuT does not assume the parameters configured manually.
- The DuT cannot be pinged from a host on another subnet with one or both payload sizes.
- The DuT does not reply with a TTL value that is > 16.

11.3 DHCP Configuration of Management Interface Test*

*Only applied to devices with out-of-band management interface(s).

Description: Tests the ability of DuT to receive the IPv4 address, subnet mask, and default gateway parameters for the management interface via DHCP. Also verifies that ICMP messages are not blocked and the TTL value of ICMP replies. The DuT will be pinged from a host in another subnet. DuT is expected to reply to ICMP echo requests (ping) coming from another subnet, while properly utilizing a default gateway, and with TTL > 16. This test will not involve an IP address change, a demonstration of basic configurability will suffice. The ICMP ping test will be done twice with 2 payloads: 32 and 56 bytes, the DuT is expected to properly reply to both.
Validation method: Console log from a device sending ICMP echo requests is used for validation.

Pass criteria:

- The DuT assumes the parameters assigned via DHCP.
- The DuT can be pinged from a host on another subnet.
- The TTL value of ICMP echo reply is > 16 for both payload sizes.

No pass criteria:

- The DuT does not assume the parameters assigned via DHCP.
- The DuT cannot be pinged from a host on another subnet with one or both payload sizes.
- The DuT does not reply with a TTL value that is > 16.

11.4 DHCP Configuration of IPMX Media Network Interface Test

Description: Tests the ability of DuT to receive the IPv4 address, subnet mask, and default gateway parameters for the IPMX Media interface via DHCP. Also verifies that ICMP messages are not blocked and the TTL value of ICMP replies. The DuT will be pinged from a host in another subnet. DuT is expected to reply to ICMP echo requests (ping) coming from another subnet, while properly utilizing a default gateway, and with TTL > 16. This test will not involve an IP address change, a demonstration of basic configurability will suffice. The ICMP ping test will be done twice with 2 payloads: 32 and 56 bytes, the DuT is expected to properly reply to both.
Validation method: Console log from a device sending ICMP echo requests is used for validation.
Pass criteria:

- The DuT assumes the parameters assigned via DHCP.
- The DuT can be pinged from a host on another subnet.
- The TTL value of ICMP echo reply is > 16 for both payload sizes.

No pass criteria:

- The DuT does not assume the parameters assigned via DHCP.
- The DuT cannot be pinged from a host on another subnet with one or both payload sizes.
- The DuT does not reply with a TTL value that is > 16.

12 Media Network Specific Tests

Description: This set of Tests validates the basic and advanced behaviors of the DuT related to PTP synchronization and multicast addressing capabilities requirements.
At the time of testing, Testers will provide to the Vendor representative sets of parameters consisting of:

- A PTP profile values compliant to TR-10-1. Any parameter values allowed by the profile may be used.
- A set of multicast addresses.
- IGMP V3 Source-specific multicast (SSM) will not be tested at this time.

The Vendor representative will be expected to configure the DuT with the given sets of parameters.

12.1 IPMX Sender Basic PTP Configuration Test

Description: The Test validates the basic PTP behavior of the DuT in follower only mode (defaultDS.slaveOnly set to TRUE).

`Note: The DuT that has a dedicated network port for PTP (not management and not an essence port) will be expected to expose the IP address configurability as per test in section 11.`

A set of PTP-related tests will be executed to test:

- The ability of the DuT to lock to the Grandmaster with a freely assignable PTP Domain number and follow the parameters of the PTP profile communicated by a Grandmaster and Master port to which the DuT is connected to: The PTP profile values may be modified in a Grandmaster, the DuT will be expected to re-lock to it.
  Validation method: Wireshark and/or a tool like PTP Track Hound for analysis.
- The ability of the DuT to be set in a follower only mode: the DuT is expected not to assume a Grandmaster role even if there is no Grandmaster present.
  Validation method: By changing the PTP domain setting on the DuT, the flow of valid PTP announce messages is interrupted, a Wireshark and/or PTP Track Hound analysis of the PTP traffic.
- The ability of the DuT to maintain proper PTP communication according to the parameters communicated by a Grandmaster: the DuT is expected to keep a stable lock to PTP and maintain the messages rate communicated by a Grandmaster or manually configured.
  Validation method: Wireshark and/or a tool like PTP Track Hound for analysis.
- The ability of the DuT not to reply to TLV management messages with multicast acknowledgment.
  Validation method: The PTP Grandmaster will be using the defined PTP profile as per TR-10-1 and will be sending the SMPTE TLV management messages once per second. Wireshark will be used to monitor from a mirrored port of the switch connected to the DuT and to verify that the BC is sending the TLV to the DuT and that the DuT while being locked to PTP, is not responding inappropriately.

Pass criteria:

- It is possible to freely assign PTP domain parameters into DuT.
- The DuT can lock to the Grandmaster, can maintain a stable lock, and maintains PTP communication according to the parameters communicated by a Grandmaster, or manually configured.
- The DuT does not assume a master role if there is no Grandmaster present.
- The DuT does not reply to TLV management messages with multicast acknowledgment.

No pass criteria:

- It is not possible to freely assign PTP domain parameters into DuT.
- The DuT cannot lock to the Grandmaster, cannot maintain a stable lock, or does not maintain PTP communication according to the parameters communicated by a Grandmaster, or manually configured.
- The DuT assumes a Grandmaster role if there is no Grandmaster present.

- The DuT replies to TLV management messages with multicast acknowledgment.

12.2 IPMX Sender Default Multicast Configuration Test

Description: The Test validates the default configuration of destination multicast IPv4 addresses and port of the DuT.

Validation method: DuT is initialized with default settings. The SDP object of the DuT is retrieved using NMOS IS-05 and is examined for compliance.

Pass criteria:

- The DuT SDP file has a destination address and port that corresponds to TR-10-9 section 17.

No pass criteria:

- The DuT SDP file has a destination address or port that does not correspond to TR-10-9 section 17.

12.3 Multicast Exclusion Range Configurability Test

Description: The Test validates the advanced configurability of multicast IPv4 addresses of the DuT. The ability to configure a given set of randomly picked multicast IP addresses compliant to TR-10-9 section 17 is checked. The range for an IPMX Receiver is 224.0.2.0 - 239.255.255.255 and for an IPMX Sender 239.0.0.0 – 239.127.255.255. In all cases the DuT shall not allow the use of 224.0.0.0 - 224.0.1.255 range.

Validation method: NMOS IS-05 will be used to set the multicast address of the DuT. Return code will be analyzed to determine if DuT accepted or refused the given multicast address.

Pass criteria:

- The DuT allows configuration of multicast addresses randomly picked as defined in the description.
- The DuT doesn't allow the use of 224.0.0.0 - 224.0.1.255 multicast address range.

No pass criteria:

- The DuT does not allow configuration of multicast addresses randomly picked as defined in the description.
- The DuT allows the use of 224.0.0.0 - 224.0.1.255 multicast address range.

13 TR-10 Sender Tests

Description: This set of tests validates the basic DuT's conformance to TR-10. These tests are performed with a PTP grandmaster present in the network and without any PTP grandmaster in the network. Test will include a subset of resolutions and formats the DuT supports. The Vendor representative is expected to provide the test administrator with a description of resolutions, number of channels and format capabilities for the DuT. The test administrator and the vendor representative will agree on the exact subset to be tested. If the DuT supports both RGB 444 8 bits and YUV 422 10 bits video format (16 bits and 24 bits for audio), a reasonable effort should be

made to test both. A minimum of two different resolution/format shall be tested so that the IPMX Info Block's Block version can be tested for compliance in Section 13.3.
For example if the DuT supports only full HD resolutions in both YUV and RGB, a minimum subset might be selected to be:

- 1920x1080p59 YUV 422 10 bits
- 1920x1080p60 RGB 8 bits

Transmitter tests for Audio at least one of the following

- 48000 Hz 16 bits PCM 1 channel, 1mSec
- 48000 Hz 16 bits PCM 2 channels, 1mSec
- 48000 Hz 16 bits PCM 8 channels, 1mSec
- 48000 Hz 24 bits PCM 1 channel, 1mSec
- 48000 Hz 24 bits PCM 2 channels, 1mSec
- 48000 Hz 24 bits PCM 8 channels, 1mSec

First steps: The Vendor representative will be expected to configure the DuT with default IPMX Multicast address and port settings. DuT is expected to be able to initiate a stream with a given set of stream parameters. Standard MTU size will always be expected. DSCP values for QoS are expected to be configured according to TR-10-9. When the test is performed with a PTP grandmaster present the vendor representative will be expected to configure the DuT accordingly.

13.1 TR-10 Sender Stream Basic Test

Description: The Vendor representative is expected to configure the DuT so that it initiates a stream of a given configuration with the default multicast address and port number. Packets of the stream are expected to have valid source and destination MAC and IPv4 addresses.
Validation method: A packet capture will be initiated before NMOS IS-05 is used to activate the stream. This guarantees that the packet capture contains the first RTCP Sender Report and corresponding media packets. The stream is analyzed; source and destination MAC and IPv4 addresses are validated. The stream parameters may be requested to be changed and a new packet capture performed and analyzed.
Pass criteria:

- DuT is capable of initiating a stream with the provided IPv4 destination multicast address and provided stream parameters.
- Stream uses a valid MAC address that corresponds to the IPv4 destination multicast address.
- Stream uses the expected DSCP values. DSCP value of the media stream are according to TR-10-9.
    - For TR-10-2 and TR-10-11 payload, DSCP = AF42(36)
    - For TR-10-3 payload, DSCP = AF41(34)

No pass criteria:

- DuT is not capable of initiating a stream with given IPv4 address parameters or provided stream parameters.
- Stream uses an invalid multicast MAC or a MAC address that does not correspond to the IPv4 destination multicast address.
- Stream does not use the expected DSCP value.

13.2 SDP object verification for a Sender

Description: While the stream initiated during test 0 is still active. NMOS IS-05 is used to retrieve the DuT SDP object. The SDP object is verified to correspond to the essence type under test. The DuT is also expected to dynamically update the SDP object if the stream parameters are changed.
Validation method: The SDP object is retrieved using NMOS IS-05 and is checked with a validation tool and/or manually. The SDP object is then compared with the required parameters listed below.
The stream parameters may be requested to be changed, and the change must be reflected in the updated SDP object.

The following parameters will be verified for all essence types:

- RTP Header parameters duplicated in the SDP:
    - c=IN <dest IP address>
    - a=source-filter (<dest IP address> <source ip address>)
    - a=fmtp:<PT>
- SSRC (if present)

With PTP:

- ts-refclk:ptp=IEEE1588 form
- mediaclk:direct:0 or mediaclk:sender

Without PTP:

- ts-refclk:localmac form
- mediaclk:direct:0 or mediaclk:sender

The following parameters will be verified for TR-10-2 and TR-10-11 essence types:

- Required Media type Parameters:
    - media clock rate (eg 90000)
    - media type (m=video)
    - width
    - height
    - exactframerate
    - sampling
    - depth

    - colorimetry
    - PM
    - SSN
    - IPMX
    - htotal
    - vtotal
    - measuredpixclk
    - interlace (if present, absent = Progressive)
- 2110-21 Required Parameter
    - TP (2110TPW or 2110TPN or 2110TPNL)

- The following parameters will be verified for TR-10-3 essence types:
    - ptime
    - media type (m=audio)
    - rtpmap: payload format (eg L24, AM824)
    - rtpmap: sampling rate (eg 48000)
    - rtpmap: channel count (eg 2)
    - fmtp: channel-order (if present)
    - fmtp: IPMX
    - fmtp: measuredsamplerate

Pass criteria:

- The DuT exposes a valid SDP object that matches the essence format.
- No warnings from the validation tools are produced.
- The SDP object is correctly updated following the change in the stream parameters.

No pass criteria:

- NMOS IS-05 can not be used to retrieve the DuT SDP object, or the SDP object is not valid.
- The SDP object is not updated following the change in the stream parameters or is updated incorrectly.

13.3 IPMX RTCP Sender Report Validation tests

Description: Validate that DuT sends valid RTCP Sender reports at the proper multicast address and port with the proper content. The Vendor representative will be asked to configure the DuT to send a specific stream format. The stream parameters may be requested to be changed, and the change must become reflected in the RTCP Sender Reports.

Validation method: PCAP captured in step 13.1 is analyzed with Wireshark using the RTCP Sender Report LUA plug-in or a IPMX test script. The first RTCP Sender report packet of the file is analyzed as it corresponds to the initial RTCP Sender report of the media stream.

Pass criteria:

- Stream uses a valid MAC address that corresponds to the IPv4 destination multicast address.
- DSCP value of the RTCP report is
  TR-10-2 and TR-10-11 payload and RTCP = AF42(36)
  TR-10-3 payload and RTCP = AF41(34)

- RTCP Sender report uses a UDP port that is +1 from the media it is associated with.
- RTCP Sender Report ‘s sender info field has a RTP timestamp value that matches the RTP timestamp of the media it is associated with.
- RTCP Sender Report ‘s sender info field has a RTP timestamp value that complies with TR-10-1 section 8.6 with respect to the NTP Timestamp field.
- When a PTP grandmaster is present during the test:
  - RTCP sender Report has NTP Timestamp fields that corresponds to PTP time
- RTCP Sender report includes a valid IPMX info block extension:
  - A Valid IPMX tag with a value of 0x5831
  - Proper Info Block length
  - Block version changes when the stream parameters are changed
  - ts-refclk string matches the SDP content
  - mediaclk string matches the SDP content
- RTCP sender Report includes a valid media info block
  - Media Info Block length is valid
  - Media info block content matches the SDP and the Stream
- For IPMX TR-10-2 and TR-10-11 Senders
  - RTCP Sender report is sent every frame/field
  - RTCP Sender Report is sent before the first media packet containing the associated RTP TS but after the first video media packet of the previous frame or field.
  - RTCP Sender Report ‘s sender info field has a RTP timestamp that increments properly. The increment corresponds to the frame rate of the media essence. 1501.5 for 59.94fps, 1800 for 50fps, 1500 for 60fps.
  - Media Info Block type is valid for DUT media stream. The value is 0x0001 for TR-10-2 and 0x0003 for TR-10-11.
- For IPMX TR-10-3 Senders
  - RTCP Sender report is sent every 10mSec
  - RTCP Sender Report is sent before the first media packet containing the associated RTP TS but after the previous audio RTCP sender Report and its associated audio media packet
  - RTCP Sender Report ‘s sender info field has a RTP timestamp that increments properly. The increment corresponds to the audio sampling rate of the media stream. 480 for 48kHz.
  - Media Info Block type is valid for DUT media stream. The value is 0x0002 for TR-10-3.

13.4 Stream validation test

Description: A stream initiated during test 0 is received with a reference receiver. A basic subjective visual or audio test is done. The stream is expected to have no obvious visual or audio artifacts of digital nature or any other Disruptions.

Validation method: A stream is visualized or listened to using a reference receiver. The stream will be under test for at least ~60 seconds.

Pass criteria:

- The stream can be received and decoded by a reference receiver.The signal is free from Disruptions.

No pass criteria:

- The stream cannot be decoded or contains Disruptions. A Vendor representative will have an opportunity to articulate why this might be ok.

13.5 TR-10-1 Sender compliance test

Description: A stream generated during test 0 is expected to be compliant to VSF TR-10-1 and TR-10-9. A stream will be analyzed with a test and measurement device to validate conformance.

Validation Method: The generated essence stream will be analyzed for good behavior according to the network compatibility model and the virtual receive buffer model of TR-10-1 section 8. The timing behavior interframe will be tested for compliance with TR-10-9 Section 11.2. The stream will be analyzed online. It can also be analyzed offline with a network capture.

Pass criteria:

- The test passes if the maximum level of the network compatibility model does not exceed the maximum as described in the TR-10-1 section 8.
- The maximum level of the virtual receive buffer does not exceed the value described in TR-10-1 section 8.
- The timing of the RTCP and First packet arrival time conforms to TR-10-9 section 11.2

No pass criteria:

- The measured value exceeds Cmax as specified in TR-10-1 section 8.
- The maximum value for the virtual receive buffer as specified in TR-10-1 section 8 is exceeded.
- The timing of the RTCP or the First packet arrival time does not conform to TR-10-9 section 11.2

14 TR-10 Receiver tests

Description: This set of Tests validate the basic receiving capabilities of the DuT.

First steps: NMOS IS-05 will be used to connect the DuT to a reference IPMX Sender. It is expected that the DuT is able to receive the signal and is expected to have no Disruptions. Standard MTU size will always be used.

For video capable devices:

The Vendor representative is expected to provide the test administrator with the resolution capabilities for the DuT. The test administrator and the vendor representative will agree on the exact subset of resolutions to be tested. Both YUV 422 10 bits and RGB 444 8 bits format will be tested.

For example, if the DuT supports only full HD resolutions, a minimum subset might be selected to be:

- 1920x1080p59 YUV 422 10 bits
- 1920x1080p60 RGB 8 bits

For audio capable devices:

The following audio format will be tested.

- 48000 Hz 16 bits PCM 1 channel, 1mSec
- 48000 Hz 16 bits PCM 2 channels, 1mSec
- 48000 Hz 16 bits PCM 8 channels, 1mSec
- 48000 Hz 24 bits PCM 1 channel, 1mSec
- 48000 Hz 24 bits PCM 2 channels, 1mSec
- 48000 Hz 24 bits PCM 8 channels, 1mSec

The audio DuT is not expected to reproduce all channels from the stream if it is not designed to do so (e.g. a stereo-output device will be expected to receive an 8-channel stream but output a minimum of selected 2 channels).

14.1 IGMPv2 test for a Receiver

Description: The Vendor representative is requested to have their DuT configured for IGMPv2 communication. IGMPv2 communication is expected between the DuT and a network switch and the DuT is expected to use a IGMPv2 group report.

Validation method: The network switch will be configured for IGMPv2 operation. NMOS IS-05 is used to connect the DuT to a reference IPMX Sender. IGMP communication is monitored via the network switch IGMP message debugging on a DuT switchport, or alternatively with Wireshark from a mirrored port of the switch connected to the DuT, if required or using a test script that monitors traffic on 224.0.0.1 for IGMPv2 Membership Report from the DuT IPv4 address.

Pass criteria:

- The DuT maintains IGMPv2 communication with a switch and successfully joins the multicast group with the (*, G) group report.

No pass criteria:

- The DuT uses another version of IGMP, cannot join the multicast group.

14.2 IGMPv3 SSM test for a Receiver

Description: The Vendor representative is requested to have their DuT configured for IGMPv3 communication. IGMPv3 communication is expected between the DuT and a switch and the DuT is expected to use a (S, G) IGMPv3 group report (from source address provided).
Validation method: The network switch will be configured for IGMPv3 operation. NMOS IS-05 is used to connect the DuT to a reference IPMX Sender. IGMP communication is monitored via switch IGMP message debugging on a DuT switchport, or alternatively with Wireshark from a mirrored port of the switch connected to the DuT, if required or using a test script that monitors traffic on 224.0.0.1 for IGMPv3 Membership Report from the DuT IPv4 address.

Pass criteria:

- The DuT maintains IGMPv3 communication with a switch and successfully joins the multicast group with the specific source address provided using (S, G) group report.

No pass criteria:

- The DuT uses another version of IGMP, cannot join the multicast group, or uses (*, G)

14.3 TR-10 Receiver Validation Tests

14.3.1 Using Baseband Reference Sender

Description: NMOS IS-05 will be used to connect the DuT to a reference Baseband IPMX Sender that the DuT is able to receive. A basic subjective visual and/or audible test is done. The stream is expected to have no Disruptions.
Validation method: A stream is received using the DuT. The vendor representative will need to demonstrate to the test administrator that the DuT is able to receive the stream without Disruptions. The stream will be observed for at least ~60 seconds.
Pass criteria:

- The DuT is capable of receiving a stream according to its capabilities.
- The video and/or audio signal is free from Disruptions.

No pass criteria:
The stream cannot be decoded or shows Disruptions. A Vendor representative will have an opportunity to articulate why this might be ok.

14.3.2 Using Synthetic Reference Sender

Description: NMOS IS-05 will be used to connect the DuT to a reference Synthetic IPMX Sender that the DuT is able to receive. A basic subjective visual and/or audible test is done. The stream is expected to have no Disruptions.

Validation method: A stream is received using the DuT. The vendor representative will need to demonstrate to the test administrator that the DuT is able to receive the stream without Disruptions. The stream will be observed for at least ~60 seconds.

Pass criteria:

- The DuT is capable of receiving a stream according to its capabilities.
- The video and/or audio signal is free from Disruptions.

No pass criteria:

The stream cannot be decoded or exhibit Disruptions. A Vendor representative will have an opportunity to articulate why this might be ok.

14.3.3 Using ST 2110 Reference Sender

Description: NMOS IS-05 will be used to connect the DuT to a reference ST 2110 Sender that the DuT is able to receive. A basic subjective visual and/or audible test is done. The stream is expected to have no Disruptions.

Validation method: A stream is received using the DuT. If the DuT is not capable of reproducing the stream, it is allowed to loopback the stream back into the network and the visual test is done on a reference device. The stream will be observed for at least ~60 seconds.

Pass criteria:

- The DuT is capable of receiving a stream according to its capabilities.
- The video and/or audio signal is free from Disruptions.

No pass criteria:

The stream cannot be decoded or shows Disruptions. A Vendor representative will have an opportunity to articulate why this might be ok.

15 NMOS Node Discovery Tests

Description: This set of tests validates a Media Node's ability to interact with an IS-04 Registry, with discovery achieved via both mDNS and unicast DNS. Tests will be carried out using the two IS-04 Registry servers with two different priorities. For unicast DNS tests a DNS server will be available on the network. During the main mDNS tests the DNS server service will be turned off. DuT will be expected to follow the behavior for registry discovery using both mDNS and unicast DNS prescribed in TR-10-8 section 7 and TR-10-9 section 15.

First steps: The management interface(s) and media interface(s) of the Device under Test (DuT) will be connected to network ports of appropriate speed and media type. Two NMOS IS-04 Registry servers will be available on the network. A high priority Registry will be set to a priority of 80-89 and the low priority Registry will be set to a priority of 90-99. A DNS server will be configured with the relevant entries for unicast DNS-SD tests.

15.1 Registry Discovery using mDNS

Description: This test validates a DuT's ability to discover, register and stay registered with an IS-04 Registry, using the mDNS discovery service. For this test, the two IS-04 Registry will only be advertised through mDNS. The Vendor representative will be expected to configure the DuT with default settings for DNS-SD browse operation as specified in TR-10-9 section 15.

Validation method: The DNS service will be turned off. The two Registry servers will be started and will advertise their presence using mDNS on the network segment the DuT uses for NMOS management. The DuT will be re-started and will be expected to find and register with one of the Registry servers on the network. A test will be done to see on which Registry server the DuT registered itself.

Pass criteria:

- The DuT registers with the high priority server.
- The DuT keeps its registration alive.

No pass criteria:

- The DuT does not register with the high priority server.
- The DuT does not keep its registration alive.

15.2 Registry Fail Over when using mDNS

Description: This test validates a DuT's ability to select a new Registry when the current Registry service it is registered with goes offline.

Validation method: This test will be executed immediately after test 15.1. The high priority Registry server will be disconnected from the network. The DuT is expected to realize the current registry it has registered with in 15.1 is no longer responsive. The DuT is expected to select a new Registry as per TR-10-8 section 7 and TR-10-9 section 15. A test will be done to see on which Registry server the DuT registered itself. The high priority Registry will then be brough back online. The DuT is expected to stay connected to the low priority Registry as per TR-10-8 section 7 and TR-10-9 section 15. A test will be done to confirm the DuT stayed registered with the low priority Registry.

Pass criteria:

- The DuT registers with the low priority Registry server when the high priority Registry server goes offline.
- The DuT stays registered with the low priority server when the high priority Registry server comes back online.
- The DuT keeps its registration alive with the low priority Registry server once registered with the low priority Registry server.

No pass criteria:

- The DuT does not register with the low priority Registry server when the high priority Registry server goes offline.
- The DuT registers with the high priority server when it comes back online.

- The DuT does not keep its registration alive with low priority server once registered with the low priority Registry server.

15.3 Registry Discovery using Unicast DNS-SD

Description: This test validates a DuT's ability to discover, register and stay registered with an IS-04 Registry, using the unicast DNS-SD discovery service. For this test, the two IS-04 Registry will only be advertised through unicast DNS-SD. The Vendor representative will be expected to configure the DuT with default settings for DNS-SD browse operation as specified in TR-10-9 section 15.

Validation method: A DNS service with the relevant entries will be turned on. The two Registry servers will be started. The DuT will be re-started and will be expected to find and register with one of the Registry servers on the network. A test will be done to see on which Registry server the DuT registered itself.

Pass criteria:

- The DuT registers with the high priority server.
- The DuT keeps its registration alive.

No pass criteria:

- The DuT does not register with the high priority server.
- The DuT does not keep its registration alive.

15.4 Registry Fail Over when using Unicast DNS-SD

Description: This test validates a DuT's ability to select a new Registry when the current Registry service it is registered with goes offline.

Validation method: This test will be executed immediately after test 15.3. The high priority Registry server will be disconnected from the network. The DuT is expected to realize the current registry it has registered with in 15.1 is no longer responsive. The DuT is expected to select a new Registry as per TR-10-8 section 7 and TR-10-9 section 15. A test will be done to see on which Registry server the DuT registered itself. The high priority Registry will then be brough back online. The DuT is expected to stay connected to the low priority Registry as per TR-10-8 section 7 and TR-10-9 section 15. A test will be done to confirm the DuT stayed registered with the low priority Registry. The Low priority server will then be disconnected from the network. The DuT is expected to then register with the high priority server.

Pass criteria:

- The DuT registers with the low priority Registry server when the high priority Registry server goes offline.
- The DuT stays registered with the low priority server when the high priority Registry server comes back online.
- The DuT registers with the high priority Registry server when the low priority Registry server goes offline.
- The DuT keeps its registration alive once registered with a Registry server.

No pass criteria:

- The DuT does not register with the low priority Registry server when the high priority Registry server goes offline.
- The DuT registers with the high priority server when it comes back online.
- The DuT does not register with the high priority Registry server when the low priority Registry server goes offline.
- The DuT does not keep its registration alive once registered with a Registry server.

15.5 Registry Discovery when both mDNS and unicast DNS are present

Description: This test validates a DuT's ability to discover, register and stay registered with an IS-04 Registry, using the unicast DNS-SD discovery service while there are mDNS service announcements. For this test, the high priority IS-04 Registry will be advertised through mDNS. A low priority Registry service will be advertised using a DNS server. The Vendor representative will be expected to configure the DuT with default settings for DNS-SD browse operation as specified in TR-10-9 section 15.

Validation method: A DNS server properly configured to advertise the low priority Registry server will be turned on and connected to the network. The high priority Registry server will advertise its presence using mDNS on the network segment the DuT uses for NMOS management. The two Registry servers will be started. The DuT will be re-started and will be expected to find and register with one of the Registry servers on the network. A test will be done to see on which Registry server the DuT registered itself.

Pass criteria:

- The DuT registers with the low priority server.
- The DuT keeps its registration alive.

No pass criteria:

- The DuT does not register with the low priority server.
- The DuT does not keep its registration alive.

15.6 Registry Fail Over when both mDNS and unicast DNS are present

Description: This test validates a DuT's ability to select a new Registry when the current Registry service it is registered with, fails.

Validation method: This test will be executed immediately after test 15.5. The low priority Registry server will be disconnected from the network. The DuT is expected to realize the current Registry sever it has registered with in 15.5 is no longer responsive. The DuT is expected to select a new Registry as per TR-10-8 section 7 and TR-10-9 section 15. This is expected to fail as the low priority Registry server is the only server with an entry in the DNS server. A test will be done to confirm the DuT is not registered with the high priority Registry server.

Pass criteria:

- The DuT does not register with any Registry server when the low priority Registry server goes offline.

No pass criteria:

- The DuT registers with the high priority Registry server when the low priority Registry server goes offline.

15.7 No Registration when the advertised registry is offline

Description: This test validates a DuT's ability to not use mDNS announcement when the unicast DNS discovered registry is offline even if mDNS advertised registry is available. For this test, the high priority IS-04 Registry will be advertised through mDNS. The low priority Registry service will be advertised using a DNS server. The Vendor representative will be expected to configure the DuT with default settings for DNS-SD browse operation as specified in TR-10-9 section 15.

Validation method: A DNS server properly configured to advertise the low priority Registry server will be turned on and connected to the network. Only the high priority Registry server will be started and will advertise its presence using mDNS on the network segment the DuT uses for NMOS management. The DuT will be re-started and will be expected not to register with any Registry server.

Pass criteria:

- The DuT remains un-registered.

No pass criteria:

- The DuT registers with the high priority server.

15.8 NMOS Node Peer-to-Peer Discovery

Description: This test validates a Media Node's ability to support NMOS APIs for discovery in cases where a Registry server is not available. DuT will be expected to follow the behavior for peer-to-peer discovery using mDNS prescribed in TR-10-8 section 7.

Validation method: No Registry server will be available on the network during this test. No DNS server containing entries for advertising a Registry service will available on the network. The DuT will be re-started and is expected to produce mDNS announcements of the required DNS-SD records as prescribed in TR-10-8 section 7.

Pass criteria:

- The DuT produces the required DNS-SD records using mDNS.

No pass criteria:

- The DuT does not produce the required DNS-SD records using mDNS.

16 Node API & Registration Behavior

Description: This set of tests validates a Media Node's ability to interact with an IS-04 Registry, with discovery achieved via unicast DNS. Tests will be carried out using the NMOS Testing Tool. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

First steps: The management interface(s) and media interface(s) of the Device under Test (DuT) will be connected to network ports of appropriate speed and media type. The DuT will be configured with a DNS server IP address and a DNS search domain, either via DHCP or via manual configuration. These DNS addresses will match those required by the NMOS Testing Tool. The testing tool will be configured to use a DNS_SD_MODE of 'unicast' and a DNS_SD_ADVERT_TIMEOUT sufficient to cover the DuT's boot time. The vendor will be asked to provide the IP address(es) and port which the IS-04 Node API is running on. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one.

The following test suites within the NMOS Testing Tool will be used to enable these tests:

- IS-04-01: IS-04 Node API

Media Nodes will be tested using the highest version of IS-04 which they support, with a minimum version of v1.3.

16.1 Node API Schema Conformance

Description: Tests that the Media Node's IS-04 Node API exposes all of the expected resources, and that each of these conforms to the JSON schema included in the specification.

Validation method: Execution of the NMOS Testing Tool, using the following tests with MAX_TEST_ITERATIONS set to 0 (the default) for full coverage:

- auto_node_1 - auto_node_16: Schema validation of GET and OPTIONS requests

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

16.2 Basic Node API Behavior

Description: Tests that a Media Node correctly implements various behavioral aspects of the IS-04 specification, including the BCP-002-01 grouping syntax.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_17: All Node resources use different UUIDs
- test_18: All Node clocks are unique, and relate to any visible Sources' clocks
- test_19: All Node interfaces are unique, and relate to any visible Senders and Receivers' ‘interface_bindings’
- test_20: Node's resources correctly signal the current protocol and IP/hostname
  Note: Warnings regarding the protocol of Node service 'href' values will be marked as a pass for this test
- test_20_01: Sender manifests use the expected Content-Type
- test_23: Senders and Receivers correctly use BCP-002-01 grouping syntax
- test_24: Periodic Sources specify a 'grain_rate'
  Note: Warnings will be marked as a fail for ST 2110 video as these are expected to be periodic
- test_24_01: Periodic Flows' 'grain_rate' is divisible by their parent Source 'grain_rate'
  Note: Warnings will be marked as a pass for this test, but including a grain_rate for periodic Flows is strongly recommended
- test_25: Receivers expose expected 'caps' for their API version
  Note: Warnings will be marked as a fail unless the device accepts all media_types (for example a generic stream analyzer)
- test_26: Source 'format' matches Flow 'format'

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the ‘PASS’ state for the test case.

16.3 Basic Registration

Description: Tests that a Media Node is capable of registering all of its advertised resources with an IS-04 Registry service. Additionally requires that Nodes maintain themselves in the Registry by using the heartbeat mechanism using the default interval of 5 seconds as specified in IS-04.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_04: Node can register a valid Node resource with the network registration service, matching its Node API self resource
- test_07: Node can register a valid Device resource with the network registration service, matching its Node API Device resource
- test_08: Node can register a valid Source resource with the network registration service, matching its Node API Source resource
- test_09: Node can register a valid Flow resource with the network registration service, matching its Node API Flow resource
- test_10: Node can register a valid Sender resource with the network registration service, matching its Node API Sender resource
- test_11: Node can register a valid Receiver resource with the network registration service, matching its Node API Receiver resource

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

16.4 Consistency of UUIDs

Description: This test validates that Media Nodes expose the same UUID values after a reboot or restart event, even when the Media Node has moved to a new network environment.

Validation method: During initial setup in the shared environment, with the consent of the unit under test, a snapshot will be taken of the UUIDs for the Node and its Devices, Senders, and Receivers. In the later isolated test, when the Media Node is on a different subnet, attached to a different Registry, the UUIDs will be compared after a reboot against the previous snapshot. In order to assist with this test, the UUID Checker tool included in the NMOS Testing Tool repository will be used.

Pass criteria:

- The UUID values for Node, Devices, Senders, and Receivers shall be the same, for elements of the Media Node which are in the same configuration.

17 Connection Management

Description: This set of tests validates a Media Node's ability to be controlled by a Controller using the IS-05 Connection API. Tests will be carried out using the NMOS Testing Tool. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

First steps: As described above, the management interface(s) and media interface(s) of the Device under Test (DuT) will be connected to network ports of appropriate speed and media type. The vendor will be asked to provide the IP address(es) and port(s) which the IS-04 Node API and IS-05 Connection API are running on. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one.

The following test suites within the NMOS Testing Tool will be used to enable these tests:

- IS-05-01: IS-05 Connection Management API (used by A.4.1 through A.4.7 below)
- IS-05-02: IS-05 Interaction with IS-04 (used by A.4.8 only below)

Media Nodes will be tested using the highest versions of IS-04 and IS-05 which they support, with a minimum of v1.3 and v1.1 respectively.

17.1 Connection API Schema Conformance

Description: Tests that the Media Node's IS-05 Connection API exposes all of the expected resources, and that each of these conforms to the JSON schema included in the specification.

Validation method: Execution of the NMOS Testing Tool, using the following tests with MAX_TEST_ITERATIONS set to 0 (the default) for full coverage:

- auto_connection_1 - auto_connection_22: Schema validation of GET and OPTIONS requests

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

17.2 Basic Connection API Behaviour

Description: Tests that the IS-05 Connection API does not expose an invalid configuration, and that violations of the advertised constraints are not permitted.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_09: All params listed in /single/senders/{senderId}/constraints/ matches /staged/ and /active/
- test_09_01: All params listed in /single/senders/{senderId}/active/ match their corresponding SDP files
- test_10: All params listed in /single/receivers/{receiverId}/constraints/ matches /staged/ and /active/
- test_11: Senders are using valid combination of parameters
- test_11_01: Sender /active parameters do not use the keyword 'auto'
- test_11_02: Patched 'auto' values are translated on '/active' endpoint for all receivers
- test_12: Receiver are using valid combination of parameters
- test_12_01: Receiver /active parameters do not use the keyword 'auto'
- test_12_02: Patched 'auto' values are translated on '/active' endpoint for all receivers
- test_13: Return of /single/senders/{senderId}/staged/ meets the schema
- test_14: Return of /single/receivers/{receiverId}/staged/ meets the schema
- test_31: Sender active response schema is valid
- test_32: Receiver active response schema is valid
- test_40: Only valid transport types for a given API version are advertised

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

17.3 Single Sender Routing

Description: Tests that a Media Node with an IPMX transmission capability can have its configuration managed via the IS-05 ‘single’ resource.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_15: Staged parameters for senders comply with constraints
- test_17: Sender patch response schema is valid
- test_19: Sender invalid patch is refused
- test_22: Receiver id on staged sender is changeable
- test_23: Sender transport parameters are changeable
- test_23_01: Senders accept a patch request with empty leg(s) in transport parameters
- test_25: Immediate activation of a sender is possible
- test_38: Number of legs matches on constraints, staged and active endpoint for senders
- test_42: Transport files use the expected Content-Type

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the ‘PASS’ state for the test case.

17.4 Single Receiver Routing

Description: Tests that a Media Node with an IPMX reception capability can have its configuration managed via the IS-05 ‘single’ resource.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_16: Staged parameters for receivers comply with constraints
- test_18: Receiver patch response schema is valid
- test_20: Receiver invalid patch is refused
- test_21: Sender id on staged receiver is changeable
- test_24: Receiver transport parameters are changeable
- test_24_01: Receivers accept a patch request with empty leg(s) in transport parameters
- test_26: Immediate activation of a receiver is possible
- test_39: Number of legs matches on constraints, staged and active endpoint for receivers

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the ‘PASS’ state for the test case.

17.5 Bulk Sender Routing

Description: Tests that a Media Node with an IPMX transmission capability can have its configuration managed via the IS-05 ‘bulk’ resource, which allows multiple Senders to be re-configured at the same time.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_34: GET on /bulk/senders returns 405
- test_36: Bulk interface can be used to change destination port on all senders

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the ‘PASS’ state for the test case.

17.6 Bulk Receiver Routing

Description: Tests that a Media Node with an IPMX reception capability can have its configuration managed via the IS-05 ‘bulk’ resource, which allows multiple Receivers to be re-configured at the same time.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_35: GET on /bulk/receivers returns 405
- test_37: Bulk interface can be used to change destination port on all receivers

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the ‘PASS’ state for the test case.

17.7 Integration with IS-04

Description: Tests that the IS-05 Connection API is correctly advertised via the IS-04 Node API, and that the Senders and Receivers advertised in IS-05 match those in IS-04. This test also ensures that when the IS-05 configuration is updated, the matching IS-04 resources are updated to reflect any changes to their state (parked/unparked) and which remote Sender or Receiver they are currently connected to.

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_01: Check that version 1.3 or greater of the Node API is available
- test_02: At least one Device is showing an IS-05 control advertisement matching the API under test
- test_03: Receivers shown in Connection API matches those shown in Node API
- test_04: Senders shown in Connection API matches those shown in Node API
- test_05: Activation of a receiver increments the version timestamp

Note: Failure related to Receiver ‘caps’ ‘version’ will be marked as a pass for this test and be indicated under the results BCP-004-01 Receiver Capabilities instead.

- test_06: Activation of a sender increments the version timestamp
- test_07: Activation of a receiver from an NMOS sender updates the IS-04 subscription
  Note: If the receiver has constraints on its acceptable stream format, for example as expressed via Receiver ‘caps’ ‘constraint_sets’, the JT-NM Tested team may configure the required parameters in the Testing Tool using the ‘SDP_PARAMETERS’ setting.
- test_09: Activation of a sender to a multicast address updates the IS-04 subscription
  Note: Failure is expected for senders that only support unicast. The JT-NM Tested results will indicate that the Device under Test is not capable.
- test_12: IS-04 interface bindings array matches length of IS-05 transport_params array
- test_13: IS-04 manifest_href matches IS-05 transportfile
- test_14: IS-05 transportfile rtpmap parameters match IS-04 Source and Flow
- test_15: IS-05 transportfile fmtp parameters match IS-04 Source and Flow
- test_16: IS-05 transportfile optional fmtp parameters match IS-04 Source and Flow
- test_17: IS-05 transportfile ts-refclk matches IS-04 Source and Node

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the ‘PASS’ state for the test case.

18 SDP Transport File and Capabilities vs. NMOS State

Description: This set of tests validates that a Media Node’s operational state is consistent with its SDP transport files and declared Capabilities. Tests will be carried out using the NMOS Testing Tool. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one. See the document “Running the IPMX Test Suite” of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- IPMX-SDP IPMX-Sdp

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_02: Test that the SDP transport file matches with the video Sender, Flow and Source of the Node
- test_03: Test that the SDP transport file matches with the audio Sender, Flow and Source of the Node
- test_04: Test that the device discovers the registry and register its Node and Device resources in it
- test_05: Test that SDP transport files can be converted to CCF capabilities and verified against sender capabilities
- test_06: Test that SDP from receiver active parameters can be converted to CCF capabilities and verified against receiver capabilities

- test_07: Test that Flow, Source, and Sender information can be converted to CCF capabilities and verified against sender capabilities
- test_08: Test that Flow, Source, and Sender capabilities from associated active sender can be converted to CCF capabilities and verified against receiver capabilities

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

19 JPEG XS

Description: This set of tests validates a Media Node's supporting the IPMX JPEG XS Profile use of BCP-006-01 NMOS With JPEG XS to express support for the JPEG XS video codec. Tests will be carried out using the NMOS Testing Tool. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- BCP-006-01 NMOS With JPEG XS

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_01: Check that version 1.3 or greater of the Node API is available
- test_02: JPEG XS Flows have the required attributes
- test_03: JPEG XS Sources have the required attributes
- test_04: JPEG XS Senders have the required attributes
- test_05: JPEG XS Sender manifests have the required parameters
- test_06: JPEG XS Receivers have the required attributes
- test_07: JPEG XS Receiver parameter constraints have valid values

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

20 Capabilities and Stream Compatibility

20.1 Receiver Capabilities

Description: This set of tests validates a Media Node's use of BCP-004-01 Receiver Capabilities to express parametric constraints on the types of streams that it is capable of consuming. Tests will be carried out using the NMOS Testing Tool. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- BCP-004-01 Receiver Capabilities

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_01: Check that version 1.3 or greater of the Node API is available
- test_02: Check Receiver Capabilities

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

20.2 Sender Capabilities

Description: This set of tests validates a Media Node's use of BCP-004-02 Sender Capabilities to express parametric constraints on the types of streams that it is capable of producing. Tests will be carried out using the NMOS Testing Tool. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- BCP-004-02 Sender Capabilities

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_01: Check that version 1.3 or greater of the Node API is available
- test_02: Check Sender Capabilities

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

20.3 IS-11 Stream Compatibility

Description: This set of tests validates a Media Node's use of IS-11 Stream Compatibility Management providing a mechanism by which to configure Sources, Flows and Senders using information from Receivers. Tests will be carried out using the NMOS Testing Tool. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one. The IP address and port of the

reference Sender must be properly configured in the UserConfig.py file used by the test suite. See the document “Running the IPMX Test Suite” of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- IS-11-01: IS-11 Stream Compatibility Management API

Validation method: Execution of the NMOS Testing Tool for all the tests of the IS-11-01 test suite at once.

Pass criteria for each test case:

- For a DuT that is a IPMX Sender with an output signal based upon the conversion of a Baseband signal. If the input interface of the DuT supports an EDID mechanism, support for the Base EDID feature of IS-11 is mandatory.
- The DuT is considered to have FAIL if the result of an auto_streamcompatibility_* test case is any value other than PASS, Could Not Test or Test Disabled.
- The DuT is considered to have FAIL if the result of a test case is any value other than PASS or Could Not Test.

- if test_01_01 is Could Not Test and test_01_04 is Could Not Test
    - NO_INPUTS are supported
- if test_03_00 is Could Not Test and test_03_01 is Could Not Test
    - NO_OUTPUTS are supported
- If test_02_01 is Could Not Test
    - NO_IS11_SENDERS
- If test_04_01 is Could Not Test
    - NO_IS11_RECEIVERS

- A DuT exposing only NMOS Senders MUST have NO_IS11_SENDERS set to false or it FAIL.
    - There must be at least one IS-11 Sender
- A DuT exposing only NMOS Receivers MUST have NO_IS11_RECEIVERS set to false or it FAIL.
    - There must be at least one IS-11 Receiver
- A DuT exposing both NMOS Senders and Receivers MUST have NO_IS11_SENDERS and NO_IS11_RECEIVERS set to false or it FAIL.
    - There must be at least one IS-11 Sender and one IS-11 Receiver

- test_00_00: At least one Device is showing an IS-11 control advertisement matching the API under test (PASS)

- There are inputs (not NO_INPUTS) and all inputs are HDMI or DisplayPort:
    - test_01_00: Verify that all connected inputs have a signal (PASS)
        - There must be at least one connected and active input
    - test_01_01: Inputs with EDID support return the Effective EDID (PASS)
    - test_01_02: Inputs with Base EDID support handle putting and deleting the Base EDID (PASS)
    - test_01_03: Inputs with Base EDID support reject an invalid EDID (PASS)
    - test_01_04: Inputs without EDID support reject requests to EDID (Could Not Test)
    - test_01_05: Inputs with Base EDID increment their version and versions of associated Senders after the Base EDID gets modified (PASS)
    - test_01_06: Effective EDID updates if Base EDID changes with 'adjust_to_caps' (PASS or Could Not Test)
        - adjust_to_caps support is optional

- There are inputs (not NO_INPUTS) and all inputs are not HDMI and not DisplayPort:
    - test_01_00: Verify that all connected inputs have a signal (PASS)
        - There must be at least one connected and active input
    - test_01_01: Inputs with EDID support return the Effective EDID (Could Not Test)
    - test_01_02: Inputs with Base EDID support handle putting and deleting the Base EDID (Could Not Test)
    - test_01_03: Inputs with Base EDID support reject an invalid EDID (Could Not Test)
    - test_01_04: Inputs without EDID support reject requests to EDID (PASS)
    - test_01_05: Inputs with Base EDID increment their version and versions of associated Senders after the Base EDID gets modified (Could Not Test)
    - test_01_06: Effective EDID updates if Base EDID changes with 'adjust_to_caps' (Could Not Test)

If the DuT supports a mix of EDID based inputs (HDMI, DisplayPort) and non-EDID based inputs (SDI) then test_01_00, test_01_01, test_01_02, test_01_03, test_01_04, test_01_05, test_01_06 must not FAIL and return either PASS or Could Not Test.

- There are IS-11 senders (not NO_IS11_SENDERS)
    - test_02_00: Reset active constraints of all senders (PASS)
    - test_02_01: Verify that the device supports the concept of IS-11 Sender (PASS)

- test_02_01_01: Verify that IS-11 Senders exist on the Node API as Senders (PASS)
- test_02_02: Verify senders (generic with/without inputs) (PASS)
- test_02_02_01: Verify that the status is "unconstrained" as per our pre-conditions (PASS)
- test_02_02_03: Verify that the sender is available in the node API, has an associated flow and is inactive (PASS)
- test_02_02_03_01: Verify that the video sender supports the minimum set of video constraints" (PASS)
- test_02_02_03_02: Verify that the audio sender supports the minimum set of audio constraints" (PASS)
- test_02_02_04_01: Verify that changing the constraints of an IS-11 sender(video) changes the version of the associated IS-04 sender. (PASS)
- test_02_02_04_02: Verify that changing the constraints of an IS-11 sender(audio) changes the version of the associated IS-04 sender. (PASS)
- test_02_02_05_01: Verify that setting no-op constraints for frame(width,height), grain_rate doesn't change the flow of a sender(video). (PASS)
- test_02_02_05_02: Verify that setting no-op constraints for sample_rate doesn't change the flow of a sender(audio). (PASS)
- test_02_02_06_01: Verify that setting no-op constraints for supported constraints doesn't change the flow of a sender(video). (PASS)
- test_02_02_06_02: Verify that setting no-op constraints for supported constraints doesn't change the flow of a sender(audio). (PASS)
- test_02_02_07_01: Verify that the device adheres to the preference of the constraint_set. (PASS)
- test_02_02_07_02: Verify that the device adheres to the preference of the constraint_set. (PASS)
- test_02_03_00: Verify senders supporting inputs (PASS or Could Not Test)
    - Supporting inputs is optional
- test_02_03_01: Verify that the input is valid (PASS or Could Not Test)
    - Supporting inputs is optional
- test_02_03_02: Verify that the input passed its test suite (PASS or Could Not Test)
    - Supporting inputs is optional
- test_02_03_03: Verify that the status is "unconstrained" as per our pre-conditions (PASS or Could Not Test)
    - Supporting inputs is optional
- test_02_03_04: Verify for inputs supporting EDID and supporting changing the base EDID (PASS or Could Not Test)
    - Supporting inputs is optional

    - test_02_03_05_01: Verify for inputs supporting EDID that the version and the effective EDID change when applying constraints (video) (PASS or Could Not Test)
        - Supporting inputs is optional
    - test_02_03_05_02: Verify for inputs supporting EDID that the version and the effective EDID change when applying constraints (audio) (PASS or Could Not Test)
        - Supporting inputs is optional

All test_02_03_* result must either all be PASS or all be Could Not Test

    - test_02_04: Verify senders not supporting inputs (PASS or Could Not Test)
        - Supporting inputs is optional
    - test_02_04_01: Verify that the status is "unconstrained" as per our pre-conditions (PASS or Could Not Test)
        - Supporting inputs is optional

All test_02_04_* result must either all be PASS or all be Could Not Test. If all test_02_03_* result are PASS then all test_02_04_* must be Could Not Test. If all test_02_03_* result are Could Not Test then all test_02_04_* must be PASS.

    - test_06_01: A sender rejects Active Constraints with unsupported Parameter Constraint URN(s) (PASS)
    - test_06_02: Putting an empty 'constraint_sets' array to Active Constraints of a sender switches its state to 'unconstrained' (PASS)
    - test_06_03: Deleting Active Constrains of a sender switches its state to 'unconstrained' (PASS)

- There are outputs (not NO_OUTPUTS) and all outputs are HDMI or DisplayPort:
    - test_03_00: Connected Outputs with EDID support return the EDID (PASS)
    - test_03_01: Disconnected Outputs with EDID support and Outputs without EDID support do not return the EDID (PASS or Could Not Test)
        - There may be some but not all disconnected outputs

- There are outputs (not NO_OUTPUTS) and all outputs are not HDMI and not DisplayPort:
    - test_03_00: Connected Outputs with EDID support return the EDID (Could Not Test)
    - test_03_01: Disconnected Outputs with EDID support and Outputs without EDID support do not return the EDID (PASS)

- There are IS-11 receivers (not NO_IS11_RECEIVERS):
    - test_04_01: Verify that the device supports the concept of IS-11 Receiver. (PASS)
    - test_04_01_01: Verify that IS-11 Receivers exist on the Node API as Receivers. (PASS)
    - test_04_02: Verify receivers (generic with/without outputs) (PASS)
    - test_04_02_01: Verify that the status is "unknown" or "non_compliant_stream" as per our pre-conditions of not being master_enabled. (PASS)
    - test_04_02_02: Verify that the Receiver supports Receiver Capabilities. (PASS)

The following tests require a reference Sender to provide a stream.

- Update IS11_REFERENCE_SENDER_NODE_API_URL and IS11_REFERENCE_SENDER_CONNECTION_API_URL of UserConfig.py (see Config.py for format).

- test_04_03: Verify receivers supporting outputs (PASS or Could Not Test)
    - Supporting outputs is optional
- test_04_03_01: Verify the status of the Receiver and the associated outputs using the reference Sender to produce the video stream consumed by the Receiver (PASS or Could Not Test)
    - The reason for “Could Not Test” MUST not be “No reference video senders found”
    - Supporting outputs is optional
- test_04_03_01_01: Verify that the status of Outputs associated with video Receivers indicates that there is a signal. (PASS or Could Not Test)
    - Supporting outputs is optional
- test_04_03_02: Verify the status of the Receiver and the associated outputs using the reference Sender to produce an audio stream consumed by the Receiver (PASS or Could Not Test)
    - The reason for “Could Not Test” MUST not be “No reference audio senders found”
    - Supporting outputs is optional
- test_04_03_02_01: Verify that the status of Outputs associated with audio Receivers indicates that there is a signal. (PASS or Could Not Test)
    - Supporting outputs is optional

All test_04_03_* result must either all be PASS or all be Could Not Test

- test_04_04: Verify receivers not supporting outputs (PASS or Could Not Test)

        - Supporting outputs is optional
    - test_04_04_01: Verify the status of the Receiver. The test requires video streaming from a Sender in order to verify the state of the Receiver. (PASS or Could Not Test)
        - The reason for “Could Not Test” MUST not be “No reference video senders found”
        - Supporting outputs is optional
    - test_04_04_02: Verify the status of the Receiver. The test requires audio streaming from a Sender in order to verify the state of the Receiver. (PASS or Could Not Test)
        - The reason for “Could Not Test” MUST not be “No reference audio senders found”
        - Supporting outputs is optional

All test_04_04_* result must either all be PASS or all be Could Not Test. If all test_04_03_* result are PASS then all test_04_04_* must be Could Not Test. If all test_04_03_* result are Could Not Test then all test_04_04_* must be PASS.

The test suite produces a result file IS-11-01s.json for Senders and IS-11-01r.json for Receivers. Those file shall be analyzed using the script is11_test_analyzer.py which analyze the test results based on the actual configuration.

For example the command line for verifying the results of Senders in an environment with HDMI inputs would be:

python is11_test_analyzer.py --with-inputs --edid-supported --with-senders --without-outputs --without-receivers IS-11-01s.json

For example the command line for verifying the results of Receivers in an environment with HDMI outputs would be:

python is11_test_analyzer.py --without-inputs --edid-supported --without-senders --with-outputs --with-receivers IS-11-01r.json

Pass criteria:

Unless noted otherwise, the is11_test_analyzer needs to indicate ‘PASS’ for the “FINAL VERDICT”.

- If an IPMX device declares in its conformance IPMX Device Declaration supporting the streaming of only one video refresh rate. It is then allowed to get the result Could Not Test for the video Sender test_02_03_05_01, otherwise it must PASS.

- If an IPMX device using the UserConfig option IS11_SOURCE_EDID_VERIFICATION=True produces the expected refresh rate in the source EDID preferred mode, it is then allowed to get the result Could Not Test for test_02_03_05_01, otherwise it must PASS.

- If an IPMX device declares in its conformance IPMX Device Declaration supporting the streaming of only one audio sample rate. It is then allowed to get the result Could Not Test for the audio Sender test_02_03_05_02, otherwise it must PASS.
- If an IPMX device using the UserConfig option IS11_SOURCE_EDID_VERIFICATION=True produces the expected sample rate in the source EDID preferred mode, it is then allowed to get the result Could Not Test for test_02_03_05_01, otherwise it must PASS.

21 Optional Capabilities

21.1 HKEP

Description: This set of tests validates a Media Node's use of BCP-005-02 NMOS With IPMX/HKEP to express support for the IPMX/HKEP technology. Tests will be carried out using the NMOS Testing Tool. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- BCP-005-02 NMOS With IPMX/HKEP

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_01: Check that version 1.3+ the Node API and version 1.1+ of the Connection API are available
- test_02: Check HKEP Senders
- test_03: Check HKEP Receivers

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.
- The vendor representative will need to demonstrate to the test administrator that the Receivers of the DuT are able to receive the stream without Disruptions.

- The vendor representative will need to demonstrate to the test administrator that the Senders of the DuT are able to transmit the stream without Disruptions.

21.2 PEP

Description: This set of tests validates a Media Node's use of BCP-005-03 NMOS With IPMX/PEP to express support for the IPMX/PEP technology. Tests will be carried out using the NMOS Testing Tool. At the beginning of a test, the DuT must actively be streaming to a reference Receiver for testing the DuT Senders, from a reference Sender for testing the DuT Receivers. A vendor may also use a Sender/Receiver it its choice instead of the reference one. See the document "Running the IPMX Test Suite" of the IPMX Testing package for usage instructions.

The following test suite within the NMOS Testing Tool will be used to enable these tests:

- BCP-005-03 NMOS With IPMX/PEP

Validation method: Execution of the NMOS Testing Tool, using the following tests:

- test_01: Check that version 1.3+ the Node API and version 1.1+ of the Connection API are available
- test_02: Check that senders transport parameters having 'ext_privacy' parameters are valid
- test_03: Check that senders transport parameters having 'ext_privacy' parameters are properly validated on activation against constraints
- test_04: Check that receivers transport parameters having 'ext_privacy' parameters are valid
- test_05: Check that receiver transport parameters having 'ext_privacy' parameters are properly validated on activation against constraints
- test_06: Check that senders ECDH private/public key is regenerated on an activation with master_enable set to false
- test_07: Check that receivers ECDH private/public key is regenerated on an activation with master_enable set to false
- test_08: Check PEP Senders
- test_09: Check PEP Receivers
- test_10: Check that senders ECDH private/public key is NOT regenerated on an activation with master_enable set to true
- test_11: Check that receivers ECDH private/public key is NOT regenerated on an activation with master_enable set to true

Pass criteria for each test case:

- Unless noted otherwise, the testing tool needs to indicate the 'PASS' state for the test case.

- The vendor representative will need to demonstrate to the test administrator that the Receivers of the DuT are able to receive the stream without Disruptions.
- The vendor representative will need to demonstrate to the test administrator that the Senders of the DuT are able to transmit the stream without Disruptions.

- When testing Senders, if test_06 returns “Could Not Test” along with the message “testing ECDH private/public keys pair regeneration require inactive senders”, the Senders must be deactivated (master_enable set to false) and the test suite executed again, this time only considering the results of test_06.

- When testing Receivers, if test_07 returns “Could Not Test” along with the message “testing ECDH private/public keys pair regeneration require inactive receivers”, the Receivers must be deactivated (master_enable set to false) and the test suite executed again, this time only considering the results of test_07.

22