ETSI TS 126 139 V16.0.0 (2020-11)

TECHNICAL SPECIFICATION

LTE;
5G;
Real-time Transport Protocol (RTP) /
RTP Control Protocol (RTCP) verification procedures
(3GPP TS 26.139 version 16.0.0 Release 16)

Reference
DTS/TSGS-0426139vg00

Keywords
5G,LTE

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

shall indicates a mandatory requirement to do something

shall not indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

should indicates a recommendation to do something

should not indicates a recommendation not to do something

may indicates permission to do something

need not indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

can indicates that something is possible

cannot indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

will indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

will not indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

might indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

1 Scope

The present document describes:

- Test cases needed to ensure an adequate level of RTP operation and RTP stream monitoring.
- Test methods capable to verify that information contained in the RTP header and in RTCP is correct and consistent with the observed characteristics of the related RTP streams:
  - Between RTP/RTCP within the scope of a single RTP stream (e.g. between an RTP stream and the corresponding RTCP reporting from the remote party, or between an RTP stream and the corresponding RTCP metadata, e.g. for sampling clock accuracy compensation between RTP sender and RTP receiver).
  - Between RTP/RTCP across RTP streams in the same RTP session (e.g. between sent and received RTP streams, or between audio RTP streams and video RTP streams).
- Requirements on what constitutes acceptable RTP/RTCP protocol field values, including RTP payload header and RTP payload length, based on the observed characteristics of the related RTP streams.
- A method for an RTP/RTCP implementation to announce on the network that it has passed the necessary tests and conforms to the new specification.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] IETF RFC 3550 (2003): "RTP: A Transport Protocol for Real-Time Applications".

[3] IETF RFC 3158 (2001): "RTP Testing Strategies".

[4] IETF RFC 3551 (2003): "RTP Profile for Audio and Video Conferences with Minimal Control".

[5] IETF RFC 3711 (2004): "The Secure Real-time Transport Protocol (SRTP)".

[6] IETF RFC 3556 (2003): "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth".

[7] IETF RFC 4585 (2006): "Extended RTP Profile for Real-time Transport Control Protocol (RTCP) – Based Feedback (RTP/AVPF)".

[8] IETF RFC 5506 (2009): "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences".

[9] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction".

[10] 3GPP TS 26.131: "Terminal acoustic characteristics for telephony; Requirements".

[11] 3GPP TS 26.132: "Speech and video telephony terminal acoustic test specification".

[12] 3GPP TS 34.229-1: "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); User Equipment (UE) conformance specification; Part 1: Protocol conformance specification".

[13] IETF RFC 3611 (2003): "RTP Control Protocol Extended Reports (RTCP XR)".

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$=$	Equal to.
$\neq$	Not equal to.
$<$	Less than.
$\leq$	Less than or equal to.
$>$	Greater than.
$\geq$	Greater than or equal to.
$+$	Addition.
$-$	Subtraction.
$\cdot$	Multiplication (when unambiguous, also by juxtaposition of expressions, e.g. $a \cdot b = ab$).
$/$	Division.
%	Modulo, e.g. a % b, where the result is the remainder from a divided by b.
Σ	Sum, e.g. $\sum_{i=1}^{n} expression$ is the sum of $expression$ over index i ranging from 1 to n.
&	Bitwise AND.
$\vert$	Bitwise OR.
$\ll$	Logical shift left, e.g. $a \ll b$, where a is shifted left b bits, shifting in 0 in least significant bit.
$\gg$	Logical shift right, e.g. $a \gg b$ where a is shifted right b bits, shifting in 0 in most significant bit.
$\in$	Belongs to.
$\forall$	For all.
$\exists$	Exists.
$\nexists$	Does not exist.
$[a..b]$	Range of values, from a to b, inclusive.
$\{expression\}$	Set of unordered, unique values created from $expression$.
$\vert expression\vert$	Absolute value of $expression$.
$\lfloor expression\rfloor$	Integer value of $expression$, the floor value, rounded towards zero.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

SUT System Under Test

4 Background

Today's 3GPP conversational and real-time services (e.g. MTSI [9] and mission critical services) all use the RTP protocol [2] or the Secure RTP protocol [5] on top of UDP/IP as media transport. (S)RTP has a companion control protocol, (S)RTCP (also in [2] and [5]), which is optional but is typically also used. In the rest of the present document, only the terms RTP/RTCP are used for brevity but should be understood to be equally applicable to SRTP/SRTCP if nothing else is said.

RTCP provides means to feedback statistical characteristics of a received RTP stream from RTP receiver to RTP sender, and to carry RTP stream metadata from RTP sender to RTP receiver, both during an active RTP session and in hold conditions where no RTP is sent. While RTP/RTCP information is designed to be useful to an ongoing real-time media session, the increased focus on automation and the consequential need for service observability and automatic performance measurements also makes it convenient and common to use RTP header and RTCP as one of the information sources to monitor the RTP streams for automation purposes. This is a very straightforward approach since it was one of the very design targets for RTP/RTCP, and both RTCP and RTP are extensible and can optionally carry various types of information. Service observability is paramount to enable any tuning or optimization to achieve a well-functioning system.

However, RTCP information has little or no end-user impact on basic, single-media communication services such as a voice-only call. The reasons for this are mainly twofold:

1) One of the intended usages of RTCP feedback reporting functionality is to allow the RTP sender to adapt its sending rate to available transport capacity since a non-acknowledged transport such as UDP/IP has no built-in congestion control, but most voice-only calls today are both low-rate and fixed-rate that has no use for adaptation; and

2) One of the other intended usages of RTCP is to provide enough metadata information to allow close time synchronization of different RTP streams, such as e.g. voice and video for a video call, but a voice-only call is single-media and has no use for such time synchronization.

Therefore, there is often no direct impact on the voice service performance if a UE or network-node implementation of the RTP stack includes incorrect (e.g. all-zero or random) data in RTCP for a voice-only call. RTCP information content only matters on service level when really making use of RTCP functionality such as for quality monitoring, somehow acting on varying transport characteristics (loss, delay, jitter), or when performing inter-media synchronization.

Also, while the call setup and modification protocol, SIP/SDP, is conformance tested in 3GPP scope (by TSG RAN WG5), no media-level tests are defined or performed there when the present document is written. RTP and RTCP are considered as media-level protocols in that conformance testing. The current overall level of RTCP implementation conformance and accuracy in 3GPP devices and in RTP/RTCP-terminating network nodes is therefore mostly unknown.

Since the information content in RTP/RTCP is often neither fundamental for the user-level experience nor explicitly tested today, there is a risk that automation, performance tuning efforts, and the application using the RTP/RTCP stack will work with incorrect data, potentially making wrong decisions that could result in worse rather than better performance.

5 Test Architectures

5.1 General

This clause describes a set of terms and possible ways to arrange the equipment used in the tests in subsequent clauses.

The "system under test" is the device or software to be tested.

The "test instrument" is the equipment used to observe RTP/RTCP output from the system under test, and in applicable cases also RTP/RTCP output from data injection, similar to e.g. a "System Simulator" (SS) in TS 34.229-1 [12]. Depending on the test architecture used, it may also act as RTP/RTCP receiver for data sent from the system under test. The test instrument also includes possibility to extract, calculate, and store information as described by the test procedures in clause 6 of the present document.

The "data injection" is the device or equipment used to generate RTP/RTCP data sent to the system under test. It may be collocated or integrated with the test instrument. For some tests and test architectures, e.g. when two systems under test are interconnected, it may be part of the system under test. Data injection may also act as RTP/RTCP receiver for data sent from the system under test, e.g. when the test instrument does not include this functionality.

In all tests, it is assumed that both systems under test, the test instrument, and the data injection are active and connected to the network before starting the test procedure.

Other test architectures than the ones suggested here may be used.

5.2 Active Test Instrument

In this test architecture, depicted in Figure 5.2-1, the test instrument is capable of both observing RTP/RTCP traffic and acting as fully functional counterpart to the system under test, including both RTP/RTCP sender and receiver.

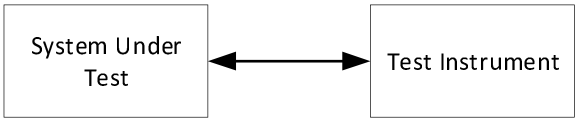

Figure 5.2-1: Active Test Instrument

5.3 Passive Test Instrument

In this test architecture, depicted in Figure 5.3-1, the test instrument is only passively observing RTP/RTCP traffic, and data injection including both RTP/RTCP sender and receiver is acting as a counterpart to the system under test.

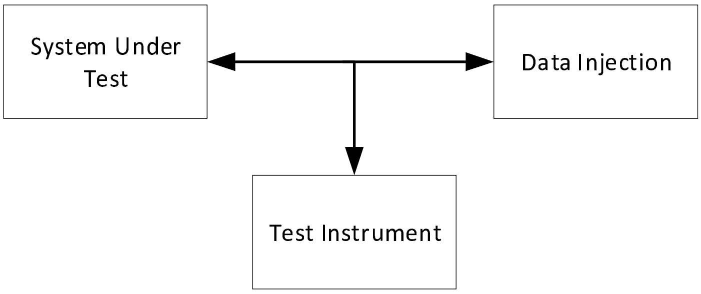

Figure 5.3-1: Passive Test Instrument

5.4 Interconnected Systems Under Test

In this test architecture, the test instrument is either only passively observing RTP/RTCP traffic, depicted in Figure 5.4-1, or actively forwarding RTP/RTCP traffic, depicted in Figure 5.4-2, while two interconnected systems under test act as each other's RTP/RTCP sender and receiver counterparts.

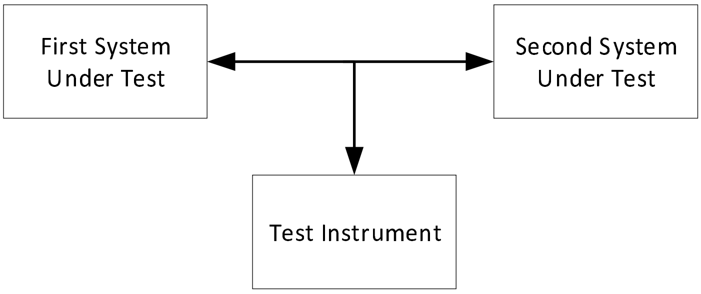

Figure 5.4-1: Interconnected Systems Under Test with Passive Test Instrument

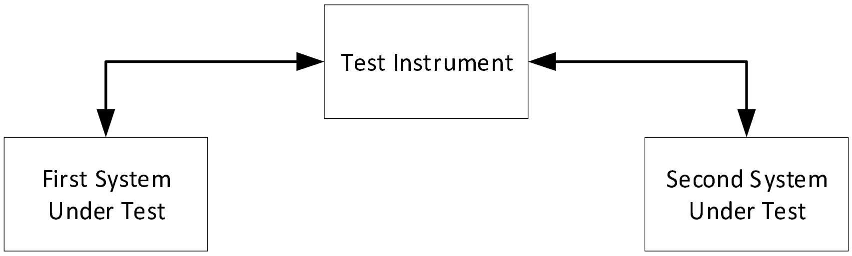

Figure 5.4-2: Interconnected Systems Under Test with Active Test Instrument

5.5 Interconnected Systems Under Test with Data Injection

In this test architecture, depicted in Figure 5.5-1, the test instrument is only passively observing RTP/RTCP traffic, while two interconnected systems under test act as each other's RTP/RTCP sender and receiver counterparts. Under the assumption that the systems under test lack necessary data injection capabilities, a separate data injection is included in the RTP/RTCP path with capability to make RTP/RTCP modifications on-path. Examples of such RTP/RTCP modifications could be setting certain sequence number or timestamp start values, dropping packets, duplicating packets, re-ordering packets, or adjusting the timing of sent packets. Individual test case procedures provide what RTP/RTCP modifications are applicable.

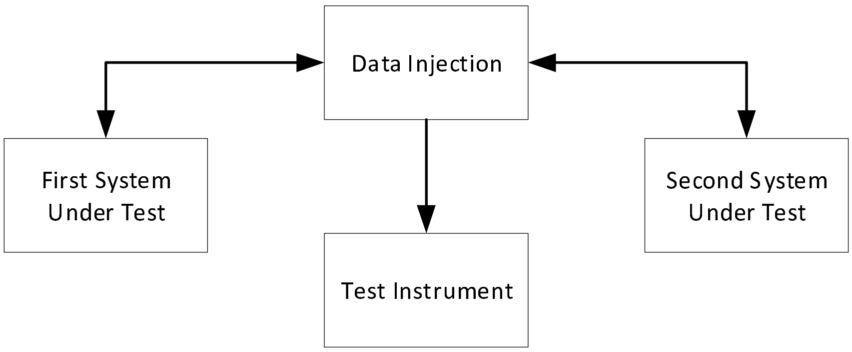

Figure 5.5-1: Interconnected Systems Under Test with Data Injection

6 Verification Tests

6.1 General

All test descriptions in this clause use the following layout:

Purpose: *Describes what is tested.*

Status: *Mandatory / Conditionally Mandatory*

Preconditions: *Describes any preconditions to be fulfilled to run the test.*

Test procedure: *Describes the steps taken to run the test.*

Stop condition: *Describes the conditions when the test is concluded, in time order if more than a single condition.*

Pass criteria: *Describes the criteria to be met to pass the test.*

Comments: *Any additional comments applicable to the test.*

Mandatory tests are required to pass for a system under test.

Conditionally Mandatory tests are mandatory to pass if the functionality is supported/implemented by the system under test, can be set or negotiated, and is correctly negotiated to be used during the test, the method of negotiation being described as part of the Preconditions section (see above).

6.2 RTCP Tests

6.2.1 General

This clause describes tests that are applicable to all types of RTCP and SRTCP packets. If not explicitly stated otherwise, any reference to RTCP is equally applicable to SRTCP. Tests that are only applicable to SRTCP and not to RTCP are specified in clause 6.4.

If not explicitly stated otherwise, all tests in this clause assume that the system under test (SUT) and the test setup are set to use:

1) AVP profile [4] for RTCP tests.
2) SAVP profile [5] for SRTCP tests.
3) A known, non-zero RTP session bandwidth.
4) Non-zero RTCP bandwidth.
5) Lossless RTP and RTCP packet transmission.
6) An RTCP report interval that is on average large enough to not exceed the set RTCP bandwidth, and where each RTCP report interval is kept within the minimum and maximum limits set by that (deterministic) average and the interval randomization described by section 6.3.1 of RFC 3550 [2].

NOTE 1: The minimum, maximum, and average RTCP report intervals are not static parameters for an RTP/RTCP implementation but, as described in RFC 3550 [2], depend on conditions that vary with e.g. RTP session bandwidth, RTCP bandwidth, average size of RTCP packets, number of RTP session participants, and use of optional RTP/RTCP features such as RTCP feedback [7] and/or non-compound RTCP packets [8].

NOTE 2: The method to set RTCP report interval in the system under test can vary. The RTCP report interval can in some cases be set directly, if the system under test has appropriate means to do so. The RTCP report interval can alternatively be set indirectly by e.g. setting an appropriate RTCP bandwidth in combination with pre-knowledge about the expected average RTCP packet size. If it is not possible to set RTCP bandwidth directly, it could be possible to set an appropriate RTP session bandwidth and rely on RTCP bandwidth becoming the default 5 % of that RTP session bandwidth.

6.2.2 Basic RTCP Tests

6.2.2.1 Initial Sending No Data RTCP Test

Purpose: Test if RTCP packets are generated by the SUT even if no RTP packets are sent or received by the SUT at the start of the session.

Status: Mandatory

Preconditions: The SUT is set to not send any RTP packets from the start of the test session, e.g. pre-configured or set as receive-only through test instrument signalling.

Test procedure: Observe the SUT output.

Stop condition: RTCP packets are sent from the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: 1) No RTP packets are sent from the SUT.
2) RTCP packets are sent from the SUT containing at least a Receiver Report.

Comments: The test result is agnostic to RTP packets being received by the SUT or not. If the SUT is not receiving any RTP packets, the Receiver Report will be empty (no report blocks, see also test 6.2.2.2).

6.2.2.2 Initial Receiving No Data RTCP Test

Purpose: Test if RTCP Sender Reports or Receiver Reports without any Receiver Report blocks are generated by the SUT even if no RTP packets are received at the start of the session.

Status: Mandatory

Preconditions: The SUT is set to not receive any RTP packets from the start of the test session, e.g. pre-configured, set as send-only through test instrument signalling, or data injection configuration.

Test procedure: Observe the SUT output.

Stop condition: RTCP packets are sent from the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: 1) No RTP packets are received by the SUT.
2) RTCP packets are sent by the SUT containing at least a Sender Report or Receiver Report, with an empty Receiver Report block, indicated by RC=0.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.2.3 Sending Data RTCP Test

Purpose: Test if RTCP Sender Reports with nonzero content are sent by the SUT when RTP packets are sent by the SUT.

Status: Mandatory

Preconditions: The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: Observe the SUT output.

Stop condition: 1) One or more RTP packets are sent by the SUT.
2) One or more RTCP packets are sent by the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: 1) RTP packets are sent by the SUT.
2) At least one RTCP packet is sent by the SUT containing at least a Sender Report.
3) The Sender Report contains nonzero Sender Info fields (NTP timestamp, RTP timestamp, sender's packet count, and sender's octet count).

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.2.4 Mid-Session Sending No Data RTCP Test

Purpose: Test if the content of Sender Reports sent by the SUT after the SUT has stopped sending RTP packets correctly reflect that RTP is no longer sent, and that RTCP transmission is changed to instead issue Receiver Reports at an appropriate time after RTP packet transmission was stopped.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.2.1 and 6.2.2.3.
2) The SUT is possible to control dynamically (e.g. directly through proprietary interfaces or via test instrument signalling) during an ongoing RTP session to turn sending of RTP packets on and off, while keeping the RTP session.

Test procedure: 1) Set the SUT to send RTP packets.
2) Observe the SUT output until at least one RTCP Sender Report is sent by the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).
3) If no RTCP Sender Report was sent by the SUT in step 2, stop the test and set it as failed.
4) Set the SUT to not send RTP packets.
5) Observe the SUT output.

Stop condition: At least three RTCP packets are sent by the SUT during step 5 in the test procedure, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1) during step 5 in the test procedure.

Pass criteria: The following conditions are fulfilled after step 5 in the test procedure:
1) No RTP packets are sent by the SUT.
2) The first two RTCP packets sent from the SUT contain Sender Reports with identical sender information in sender's packet count and sender's octet count.
3) The third RTCP packet sent from the SUT contains a Receiver Report.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.
IETF RFC 3550 [2] section 6.4 specifies to send a Sender Report if RTP was sent during this RTCP reporting interval or the previous one, otherwise a Receiver Report is sent. The two first Sender Reports in this test contain identical sender information in sender's packet count and sender's octet count since no packets were sent during the latter reporting period.

6.2.2.5 Mid-Session Receiving No Data RTCP Test

Purpose: Test if RTCP content from the SUT correctly indicates that no RTP packets were received by the SUT during the reporting period, even if RTP packets were received by the SUT previously in the RTP session.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.2.2 and 6.2.2.3.
2) The data injection to the SUT is possible to control dynamically (e.g. directly through proprietary interfaces or via test instrument signalling) during an ongoing RTP session to turn sending of RTP packets on and off, while keeping the RTP session.

Test procedure: 1) Set the data injection to send RTP packets to the SUT.
2) Observe the SUT output until at least one RTCP packet is sent from the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).
3) If no RTCP packet was sent by the SUT in step 2, stop the test and set it as failed.
4) Set the data injection to not send RTP packets to the SUT.
5) Observe the SUT output.

Stop condition: At least two RTCP packets are sent by the SUT during step 5 in the test procedure, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1) during step 5 in the test procedure.

Pass criteria: The second RTCP packet sent from the SUT during step 5 of the test procedure contains a Sender Report or Receiver Report with an empty Receiver Report block, indicated by RC=0.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.
If the data injection does not perform step 4 fast enough after step 2, the SUT may receive some RTP packets after step 2. Those RTP packets will then be reported on in the next RTCP packet sent from the SUT, which is the reason to use the second RTCP packet sent after step 4 as input to the pass criteria.

6.2.2.6 Compound RTCP Packet Format Test

Purpose: Test if the content and formatting of a compound RTCP packet sent by the SUT fulfils basic criteria.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.2.1 – 6.2.2.5.
2) Reduced-size RTCP [8] is *not* negotiated to be used in the session (see clause 6.2.10 for reduced-size tests), e.g. indicated by "a=rtcp-rsize" *not* being present in the SDP answer for the tested RTP media.

Test procedure: Observe the SUT output from tests 6.2.2.1 – 6.2.2.5.

Stop condition: Not applicable (implicit through tests in 6.2.2.1 – 6.2.2.5).

Pass criteria: All RTCP packets sent from the SUT fulfils:
1) A Sender Report or Receiver Report (equally acceptable for the outcome of this test) is included as the *first part* of the compound RTCP packet.
2) An SDES packet with a CNAME SDES item (see also test 6.2.5.2) is included in the compound RTCP packet (except when a compound RTCP packet is split for partial encryption, if used; see SRTCP tests in clause 6.4).
3) Each RTCP packet in the compound RTCP packet starts on a 32-bit boundary.
4) Each RTCP packet in the compound RTCP packet has a length field with a value that is the number of 32-bit words minus one occupied by that RTCP packet, including the RTCP packet header and any padding.
5) The UDP packet that contains the compound RTCP packet in criteria 4 as payload has a Length field with a value that is 4 times the sum of the length fields in all the RTCP packets in the compound RTCP packet (the total RTCP size in bytes), plus 8 (the UDP header size in bytes).

6.2.2.7 RTCP Report Count Test

Purpose: Test if RTCP packets sent by the SUT contain a correct and consistent report count and length fields.

Status: Mandatory

Preconditions: The SUT has passed tests 6.2.2.1 – 6.2.2.5.

Test procedure: Observe the SUT output from tests 6.2.2.1 – 6.2.2.5.

Stop condition: Not applicable (implicit through tests in 6.2.2.1 – 6.2.2.5).

Pass criteria: All RTCP packets sent from the SUT fulfils:
1) Regardless if the RTCP packet is a Sender Report or Receiver Report, the report count (RC) field is consistent with the number of report blocks observed to be included in the packet.
2) The RTCP packet length field has a value that is the number of 32-bit words minus one occupied by that RTCP packet, including the RTCP packet header and any padding.

Comments: This test does not cover a higher report count than 1, which can be required when more than a single RTP stream is sent in an RTP session, e.g. in multi-stream scenarios as described by TS 26.114 [9] Annex S. Tests covering a higher report count than 1 may be added in future revisions of the present document.

6.2.3 RTCP Bandwidth and Transmission Timing Tests

6.2.3.1 RTCP Disable Test

Purpose: Test if setting RTCP bandwidth to zero in SDP disables RTCP transmission from the SUT.

Status: Conditionally Mandatory (Precondition 1)

Preconditions: 1) RTCP transmission by the SUT is disabled for the RTP session, e.g. pre-configured or negotiated in SDP by both b=RS:0 and b=RR:0 being present in the SDP answer (zero RTCP bandwidth) through test instrument signalling.

Test procedure: Observe the SUT output.

Stop condition: Time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: No RTCP packets are sent from the SUT.

Comments: The test result is agnostic to RTP packets being sent or received by the SUT.

6.2.3.2 RTCP Basic Interval Test

Purpose: Test if the time interval with which RTCP packets are sent from the SUT keeps within the maximum RTCP bandwidth and complies with IETF RFC 3550 [2], including prescribed randomization of the time interval (if used).

Status: Mandatory

Preconditions: 1) The SUT is set to not use any explicit RTCP bandwidth such that default RTCP bandwidth is used (based on 5 % of the RTP session bandwidth), e.g. pre-configured or through neither "b=RS" nor "b=RR" (see also test 6.2.3.3) being included in the SDP answer through test instrument signalling.
2) The SUT is set to send and receive RTP packets during the test session, e.g. pre-configured or set as send-receive in SDP answer through test instrument signalling.
3) The test RTP session contains no more than two participants (a point-to-point session).

Test procedure: 1) Observe the SUT output.
2) Note the reception time (t_i) and size (S_i) in bits including UDP and IP headers of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets, n, sent from the SUT.
4) For each pair of subsequent RTCP packets, calculate the difference in send time (called the interval), $I_i = t_i - t_{i-1}, i\epsilon[2..n]$.
5) Calculate the average RTCP interval $I = \frac{1}{n-1}\sum_{i=2}^{n} I_i$.
6) Calculate the maximum RTCP interval $I_{max} = \max_i I_i$.
7) Calculate the minimum RTCP interval $I_{min} = \min_i I_i$.
8) Calculate the average RTCP bandwidth $B = (\sum_{i=1}^{n-1} S_i)/(\sum_{i=2}^{n} I_i)$.
9) Calculate the average RTCP packet size $S = \frac{1}{n-1}\sum_{i=1}^{n-1} S_i$.
10) Calculate the maximum RTCP bandwidth $B_M = 0.05B_{RTP}$, where B_{RTP} = RTP session maximum bandwidth in bit/s, e.g. calculated from b=AS (converting from kbit/s to bit/s) in SDP answer.

Stop condition: At least 10 minutes, or 100 RTCP packets are sent from the SUT, whichever time is less.

Pass criteria: 1) $B \leq B_M$.
2) $I_{min} \geq \frac{S}{(e-1.5)B_M}$.
3) $I_{max} \leq \max\left(\frac{3S}{(e-1.5)B_M} \middle| \frac{15}{2(e-1.5)}\right)$.

Comments: This test allows an exception to IETF RFC 3550 [2] and the RTP testing strategies in [3] stating that an implementation fails the test if the RTCP packets are sent with a constant interval (not randomized).
The calculations of bandwidth limits are seemingly not in accordance with the definition of IETF RFC 4566 section 5.8 that says b=AS is the RTP session bandwidth (all RTP streams that are sent and received in the RTP session), but 3GPP specifications commonly interpret SDP b= lines as receive bandwidth seen from the part sending the SDP, which is thus only half of the actual RTP session bandwidth in a bandwidth-symmetric point-to-point session.
The maximum and minimum interval limits are chosen based on the $[0.5T..1.5T]$ uniform distribution randomization of the RTCP interval, divided by the e-1.5 (=1.21828…) reconsideration factor, both taken from section 6.2.1 of IETF RFC 3550. The maximum allowed interval (the lowest RTCP bandwidth) in this test is a uniformly distributed and randomized interval based on the longest of the

interval calculated from actual RTCP packet sizes and maximum RTCP bandwidth, and the default 5 second interval.

6.2.3.3 RTCP Explicit Bandwidth Test

Purpose: Test if the SUT keeps RTCP bandwidth within set maximum limit, e.g. by "b=RS" and "b=RR" in SDP, and the information in those is correctly interpreted with respect to SUT being sender in a point-to-point RTP session.

Status: Conditionally Mandatory (Precondition 1)

Preconditions:
1) The SUT can set explicit RTCP bandwidth.
2) The SUT has passed test 6.2.3.2.
3) The SUT is set to send and receive RTP packets during the test session, e.g. pre-configured or set as send-receive in SDP answer through test instrument signalling.

Test procedure:
1) Negotiate an explicit RTCP bandwidth in SDP, setting "b=RS:625" and "b=RR:1875" in SDP answer.
2) Observe the SUT output.
3) Note the reception time (t_i) and size (S_i) in bits of each RTCP packet sent from the SUT, including UDP and IP headers.
4) Count the number of RTCP packets, n, sent from the SUT.
5) For each pair of subsequent RTCP packets, calculate the difference in send time, $I_i = t_i - t_{i-1}, i\epsilon[2..n]$.
6) Calculate the average RTCP bandwidth $B = (\sum_{i=1}^{n-1} S_i)/(\sum_{i=2}^{n} I_i)$.
7) Note the test result from step 6.
8) Negotiate an explicit RTCP bandwidth in SDP, setting "b=RS:0" and "b=RR:2500" in SDP answer.
9) Re-run steps 2-6 and note the test result from step 6.

Stop condition: At least 10 minutes, or 100 RTCP packets are sent from the SUT, whichever time is less, in each of steps 7 and 9.

Pass criteria:
1)$B \leq 2500\ bit/s$, from step 7 of this test.
2)$B \leq 2500\ bit/s$, from step 9 of this test.

Comments: Pass criteria 1 and 2 are identical because the proportion of senders in this point-to-point RTP session (100 %) is larger than RS/(RS+RR) and all RTP senders get their proportion of the (RS+RR) sum (see section 2 of IETF RFC 3556 [6]).
The calculations of bandwidth limits are seemingly not in accordance with the definition of IETF RFC 3556 section 2 that says "RS" indicates the RTCP bandwidth allocated to active data senders and "RR" indicates the RTCP bandwidth allocated to other participants in the RTP session (all RTP streams that are sent and received in the RTP session), but 3GPP specifications commonly interpret SDP b= lines as receive bandwidth seen from the part sending the SDP, which is thus only half of the actual RTP session bandwidth in a bandwidth-symmetric point-to-point session.

6.2.3.4 RTCP Timeout Test

Purpose: Test if an RTP stream received by the SUT that is stopped at the sender without any explicit RTCP BYE times out correctly in the SUT and is reflected as such in the subsequent receiver reporting from the SUT.

Status: Conditionally Mandatory (Precondition 1)

Preconditions:
1) The SUT is capable of using multiple simultaneous RTP streams in a single RTP session.
2) The SUT has passed tests 6.2.2.1, 6.2.2.3 and 6.2.2.5.
3) The data injection to the SUT is possible to control dynamically during an ongoing RTP session to turn sending of RTP packets on and off, without any indication in session signalling and while keeping the RTP session.

Test procedure:
1) Set the data injection to send RTP packets to the SUT.
2) Observe the SUT output until at least one RTCP packet is sent from the SUT.
3) Set the data injection to not send RTP packets to the SUT, not sending any RTCP BYE for the

stopped RTP stream SSRC.
4) Observe the SUT output.

Stop condition: Six (6) RTCP packets are sent from the SUT after step 3 in the test procedure.

Pass criteria: The 6th RTCP packet sent from the SUT contains a Sender Report or Receiver Report that does *not* contain any report block for the stopped SSRC from data injection.

Comments: An SSRC for which no packets have been received during (per default) 5 or more RTCP intervals will be timed out, as described by section 6.3.5 of IETF RFC 3550 [2]. The stopped SSRC may be omitted in RTCP reporting from the SUT (but not timed out) already for the first RTCP reporting interval where no RTP packets for that SSRC reaches the SUT, as described by section 6.4 of IETF RFC 3550. The current test procedure and pass criteria do not test if RTCP bandwidth allocated to the SSRC that was timed out is correctly re-allocated to the remaining SSRCs, which may be added in a subsequent version of the present document.

6.2.3.5 RTCP Step Join Backoff Receiver Test

Purpose: Test if the time between first and second RTCP packet sent from the SUT is appropriate (using sufficiently low but not too low RTCP bandwidth) when the SUT is joining as RTP receiver (non-sender) in an RTP session that the SUT can discover to already contain many participants, as described by section 6.2 of IETF RFC 3550 [2] and section 2.4.2 of IETF RFC 3158 [3].

For Further Study.

6.2.3.6 RTCP Step Join Backoff Sender Test

Purpose: Test if the time between first and second RTCP packet sent from the SUT is appropriate (using sufficiently low but not too low RTCP bandwidth) when the SUT is joining as RTP sender in an RTP session that the SUT can discover to already contain many participants, as described by section 6.2 of IETF RFC 3550 [2] and section 2.4.2 of IETF RFC 3158 [3].

For Further Study.

6.2.4 Sender Report Sender Info Tests

6.2.4.1 Sender SSRC Test

Purpose: Test if all SSRC values included in RTCP sender information sent from the SUT corresponds to the SSRC of an RTP stream sent from the SUT, and that there are corresponding RTCP sender information for all RTP streams (one or more) sent from the SUT.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.2.3.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The number of simultaneously used SSRC identifiers to be sent by the SUT in the RTP session is known a-priori (only one in the single-stream case).

Test procedure: Observe the SUT output.

Stop condition: 1) RTP packets are sent by the SUT.
2) At least three RTCP packets are sent per used SSRC identifier from the SUT (at least the first RTCP packet may be sent before first RTP packet for each SSRC).

Pass criteria: 1) The "SSRC of sender" header field in all RTCP packets sent by the SUT contains a value that is identical to the "synchronization source (SSRC) identifier" header field in at least one of the RTP packets sent by the SUT.
2) The "synchronization source (SSRC) identifier" header field in all RTP packets sent by the SUT

contains a value that is identical to the "SSRC of sender" header field in at least one of the RTCP packets sent by the SUT.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.4.2 NTP Test

Purpose: Test if the NTP timestamp field in successive RTCP packets sent from the SUT progresses with a consistent clock rate, and that the used NTP time format is correct (seconds in the high 32 bits).

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.2.3.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The "NTP timestamp" field in the RTCP packet Sender Information from the SUT is monotonously increasing (value is not wrapping around, see also test 6.2.4.3).
4) The test network and test equipment contributes with an amount of time jitter that is insignificant to the test result.

Test procedure: 1) Observe the SUT output.
2) Note the reception time (t_i) and NTP timestamp field value (w_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets, n, sent from the SUT.
4) Using the first and last RTCP packets sent from the SUT, calculate the difference in reception time, $I = t_n - t_1$, and difference in NTP time, $N = w_n - w_1$.
5) Calculate the NTP timestamp clock rate estimate $C = {}^{N}/_{I}$.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least 30 seconds have passed from the first RTCP packet was sent from the SUT.

Pass criteria: $0.999 \leq {}^{C}/_{2^{32}} \leq 1.001$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.
NTP time calculations require 64-bit integer arithmetic capability.
The pass criterion corresponds to a minimum timestamp clock accuracy of 0.1 % but a much higher accuracy should in general be expected. The test setup can introduce packet delay jitter impacting measurement accuracy, which is mitigated by using a rather long test procedure and the above values are chosen as a trade-off.

6.2.4.3 Wrapped NTP Test

Purpose: Test if the NTP timestamp field in successive RTCP packets sent from the SUT progresses with a consistent clock rate, even when the NTP timestamp field wraps around.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed test 6.2.4.2.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The "NTP timestamp" field in the first RTCP packet Sender Information sent from the SUT is set to a value close to 2^{64}-1 (when interpreted as an unsigned 64-bit integer), such that the value will wrap during the test.
4) The test network and test equipment contributes with an amount of time jitter that is insignificant to the test result.

Test procedure: 1) Observe the SUT output.
2) Note the reception time (t_i) and NTP timestamp field value (w_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets, n, sent from the SUT.
4) Using the first and last RTCP packets sent from the SUT, calculate the difference in reception time, $I = t_n - t_1$.

5) Using the first and last RTCP packets with $w_n < w_1$ when interpreting w_n and w_1 as unsigned 64-bit values, calculate the difference in NTP time $N = 2^{64} - w_1 + w_n = (2^{64} + w_n - w_1)\ \%\ 2^{64}$.
6) Calculate the NTP timestamp clock rate estimate $C = {}^{N}/_{I}$.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least 30 seconds have passed from the first RTCP packet was sent from the SUT.
3) The "NTP timestamp" field in the last RTCP packet Sender Information sent from the SUT is less than the "NTP timestamp" field in the first RTCP packet Sender Information sent from the SUT, when interpreting those fields as unsigned 64-bit integers.

Pass criteria: $0.999 \leq {}^{C}/_{2^{32}} \leq 1.001$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.
NTP time calculations require 64-bit integer arithmetic capability.
The pass criterion corresponds to a minimum timestamp clock accuracy of 0.1 % but a much higher accuracy should in general be expected. The test setup can introduce packet delay jitter impacting measurement accuracy, which is mitigated by using a rather long test procedure and the above values are chosen as a trade-off.

6.2.4.4 Time Stamp Test

Purpose: Test if the RTP timestamp field in successive RTCP packets sent from the SUT progresses with a clock rate that, in relation to the NTP timestamp field, is reasonably close to the set rate for the RTP payload type related to this SSRC, e.g. from "a=rtpmap" line in SDP.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.2.3, 6.2.4.1 and 6.2.4.2.
2) The SUT is set to send RTP packets for a single RTP stream (one SSRC) during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The "RTP timestamp" field in the RTCP packet Sender Information sent from the SUT is monotonously increasing (value is not wrapping around, see also test 6.2.4.5).

Test procedure: 1) Observe the SUT output.
2) Note the NTP timestamp field value (w_i) and RTP timestamp field value (r_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets, n, sent from the SUT.
4) Using the first and last RTCP packet sent from the SUT, calculate the difference in NTP time, $N = w_n - w_1$ and difference in RTP timestamp, $T = r_n - r_1$.
5) Calculate the NTP timestamp difference in seconds $D = {}^{N}/_{2^{32}}$.
6) Calculate the RTP timestamp rate estimate $R = {}^{T}/_{D}$.
7) Retrieve RTP Time Stamp rate information, S, e.g. from "<clock rate>" on the SDP "a=rtpmap:<payload type> <encoding name>/<clock rate>" line, applicable for the RTP stream (SSRC) sent from the SUT that the RTCP packets sent from the SUT are associated with.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least 30 seconds have passed from the first RTCP packet was sent from the SUT.

Pass criteria: $0.999 \leq {}^{R}/_{S} \leq 1.001$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.
NTP time calculations require 64-bit integer arithmetic capability.
The pass criteria correspond to a minimum timestamp clock accuracy of 0.1 % in relation to NTP time but a much higher accuracy should in general be expected.

6.2.4.5 Wrapped Time Stamp Test

Purpose: Test if the RTP timestamp field in successive RTCP packets sent from the SUT progresses with a clock rate that, in relation to the NTP timestamp field, is reasonably close to the set rate, e.g. from "a=rtpmap" line in SDP, even when the RTP timestamp field wraps around.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed test 6.2.4.4.
2) The SUT is set to send RTP packets for a single RTP stream (one SSRC) during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The "RTP timestamp" field in the first RTCP packet Sender Information sent from the SUT is set to a value close to 2^{32}-1 (when interpreted as an unsigned 32-bit integer), such that the value will wrap during the test.

Test procedure: 1) Observe the SUT output.
2) Note the NTP timestamp field value (w_i) and RTP timestamp field value (r_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets, n, sent from the SUT.
4) Using the first and last RTCP packets sent from the SUT, calculate the difference in NTP time, $N = w_n - w_1$.
5) Calculate the NTP timestamp difference in seconds $D = {}^{N}/_{2^{32}}$.
6) Using the first and last RTCP packets with $r_n < r_1$ when interpreting r_n and r_1 as unsigned 32-bit values, calculate $T = 2^{32} - r_1 + r_n = (2^{32} + r_n - r_1)\ \%\ 2^{32}$.
7) Calculate the RTP timestamp rate estimate $R = {}^{T}/_{D}$.
8) Retrieve the intended RTP Time Stamp rate information, *S*, e.g. from "<clock rate>" on the SDP "a=rtpmap:<payload type> <encoding name>/<clock rate>" line, applicable for the RTP stream (SSRC) sent from the SUT that the RTCP packets sent from the SUT are associated with.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least 30 seconds have passed from the first RTCP packet was sent from the SUT.
3) The "RTP timestamp" field in the last RTCP packet Sender Information sent from the SUT is less than the "RTP timestamp" field in the first RTCP packet Sender Information sent from the SUT, when interpreting those fields as unsigned 32-bit integers.

Pass criteria: $0.999 \leq {}^{R}/_{S} \leq 1.001$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.
NTP time calculations require 64-bit integer arithmetic capability.
The pass criteria correspond to a minimum timestamp clock accuracy of 0.1 % in relation to NTP time but a much higher accuracy should in general be expected.

6.2.4.6 Packet Count Test

Purpose: Test if the reported packet count in RTCP sent from the SUT corresponds with RTP packets sent from the SUT.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.2.3.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe the SUT output.
2) Note the "sender's packet count" field (p_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count and note the number of RTP packets (P_i) sent from the SUT between each pair $\{i-1, i\}$ of subsequent RTCP packets sent from the SUT.
5) For each pair of subsequent RTCP packets, calculate the difference in packet count, $d_i = p_i - p_{i-1}, i\epsilon[2..n]$.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least three RTCP packets are sent from the SUT after the first RTP packet sent from the SUT (one or more RTCP packets may be sent before first RTP packet).

Pass criteria: $d_i = P_i, for\ all\ i \in [2..n]$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.4.7 Wrapped Packet Count Test

Purpose: Test if the reported packet count in RTCP sent from the SUT corresponds with RTP packets sent from the SUT, even when the "sender's packet count" field wraps around.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed test 6.2.4.6.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The "sender's packet count" field in the first RTCP packet Sender Information sent from the SUT is set to a value close to 2^{32}-1 (when interpreted as an unsigned 32-bit integer), such that the value will wrap during the test.

Test procedure: 1) Observe the SUT output.
2) Note the "sender's packet count" field (p_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count and note the number of RTP packets (P_i) sent from the SUT between each pair $\{i-1, i\}$ of subsequent RTCP packets sent from the SUT.
5) For each pair of subsequent RTCP packets where $p_i \geq p_{i-1}$ when interpreting p_i and p_{i-1} as unsigned 32-bit values, calculate the difference in packet count, $d_i = p_i - p_{i-1}, i \epsilon [2..n]$.
6) For each pair of subsequent RTCP packets where $p_i < p_{i-1}$ when interpreting p_i and p_{i-1} as unsigned 32-bit values, calculate $d_i = 2^{32} - p_{i-1} + p_i = (2^{32} + p_i - p_{i-1})\ \%\ 2^{32}, i \in [2..n]$.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least three RTCP packets are sent from the SUT after the first RTP packet (one or more RTCP packets may be sent before first RTP packet).
3) The "sender's packet count" field in the last RTCP packet Sender Information sent from the SUT is less than the "sender's packet count" field in the first RTCP packet Sender Information, when interpreting those fields as unsigned 32-bit integers.

Pass criteria: $d_i = P_i, for\ all\ i \in [2..n]$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.4.8 Octet Count Test

Purpose: Test if the reported octet count in RTCP sent from the SUT corresponds with RTP packets sent from the SUT.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.2.3.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe the SUT output.
2) Note the "sender's octet count" field (c_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count and note the number of RTP packets (P_i) sent from the SUT between each pair $\{i-1, i\}$ of subsequent RTCP packets sent from the SUT.
5) Note the RTP payload size (s_{N_i}), with $N_i \in [1..P_i]$, excluding RTP, lower layer headers, and any RTP padding of each RTP packet sent from the SUT between each pair of subsequent RTCP packets sent from the SUT.
6) For each pair of subsequent RTCP packets, calculate the difference in octet count, $b_i = c_i - c_{i-1}$, and sum of RTP payload sizes $S_i = \sum_{N_i=1}^{P_i} s_{N_i}$ with $i \in [2..n]$.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least three RTCP packets are sent from the SUT after the first RTP packet sent from the SUT (one or more RTCP packets may be sent before first RTP packet).

Pass criteria: $b_i = S_i, for\ all\ i \in [2..n]$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.4.9 Wrapped Octet Count Test

Purpose: Test if the reported octet count in RTCP sent from the SUT corresponds with RTP packets sent from the SUT, even when the "sender's octet count" field wraps around.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed test 6.2.4.8.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.
3) The "sender's octet count" field in the first RTCP packet Sender Information sent from the SUT is set to a value close to 2^{32}-1 (when interpreted as an unsigned 32-bit integer), such that the value will wrap during the test.

Test procedure: 1) Observe the SUT output.
2) Note the "sender's octet count" field (c_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count and note the number of RTP packets (P_i) sent from the SUT between each pair $\{i-1, i\}$ of subsequent RTCP packets sent from the SUT,
5) Note the RTP payload size (s_{N_i}), with $N_i \in [1..P_i]$, excluding RTP, lower layer headers, and any RTP padding of each RTP packet sent from the SUT between each pair of subsequent RTCP packets sent from the SUT.
6) For each pair of subsequent RTCP packets, calculate the sum of RTP payload sizes $S_i = \sum_{N_i=1}^{P_i} s_{N_i}$ with $i \in [2..n]$.
7) For each pair of subsequent RTCP packets where $c_i \geq c_{i-1}$ when interpreting c_i and c_{i-1} as unsigned 32-bit values, calculate the difference in octet count, $b_i = c_i - c_{i-1}, i \epsilon [2..n]$.
8) For each pair of subsequent RTCP packets where $c_i < c_{i-1}$ when interpreting c_i and c_{i-1} as unsigned 32-bit values, calculate $b_i = 2^{32} - c_{i-1} + c_i = (2^{32} + c_i - c_{i-1})\ \%\ 2^{32}, i \in [2..n]$.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least three RTCP packets are sent from the SUT after the first RTP packet sent from the SUT (one or more RTCP packets may be sent before first RTP packet).
3) The "sender's octet count" field in the last RTCP packet Sender Information sent from the SUT is less than the "sender's octet count" field in the first RTCP packet Sender Information, when interpreting those fields as unsigned 32-bit integers

Pass criteria: $b_i = S_i, for\ all\ i \in [2..n]$.

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.4.10 Synchronization Information Test

Purpose: Test if the RTCP sender information sent from the SUT and corresponding RTP for two or more streams that are sent simultaneously from the SUT and that are intended to be synchronized, are consistent enough to allow RTP receiver-side synchronization.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.4.2, 6.2.4.4, and 6.2.5.2.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe SUT output.
2) Note the "SSRC of sender" field (s_i), NTP timestamp field (w_i), RTP timestamp field (r_i), and CNAME SDES item field (c_i) values of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Note the reception time (t_j), "SSRC identifier" field (S_j), and RTP timestamp field (T_j) values of each RTP packet sent from the SUT.
5) Count the number of RTP packets (N) sent from the SUT.
6) For each SSRC value to be tested (indexed with k), retrieve RTP Time Stamp rate information

(R_k), e.g. from "<clock rate>" on the SDP "a=rtpmap:<payload type> <encoding name>/<clock rate>" line, applicable for the SSRC sent from the SUT that the RTCP packets sent from the SUT are associated with ($s_i = S_j$).
7) For at least one RTCP and RTP packet pair for each SSRC value to be tested (indexed with k), calculate the nominal playout time $C_k = {}^{w_k}/_{2^{32}} + {}^{(T_k - r_k)}/_{R_k}$, where w_k, r_k, s_k, and c_k belong to a single RTCP packet, and T_k is taken from a corresponding RTP packet with $S_k = s_k$.

Stop condition: 1) At least one RTP packet from all RTP streams (different SSRC identifiers, S_j) to be synchronized are sent from the SUT.
2) At least one RTCP packet are sent from the SUT per used SSRC identifier, i.e. there exist both s_i and S_j with all SSRC values that are to be included in the test (the first RTCP packet may be sent before first RTP packet for each SSRC).

Pass criteria: The following is fulfilled for each SSRC value (s_k) to be tested, used by RTCP and RTP packets sent from the SUT:
1) The CNAME SDES item field values (c_k) are identical (if not, they are not intended to be synchronized).
2) The nominal playout times (C_k) of an RTP packet using the tested SSRC value (s_k) and an RTP packet using a different tested SSRC value (s_{k+1}) that are received no more than 500 ms apart ($|(t_{k+1} - t_k)| \leq 0.5$), are no more than 300 ms apart when adjusted for the difference in reception time; $|(C_{k+1} - C_k) - (t_{k+1} - t_k)| \leq 0.3$.

Comments: This test only verifies that RTP streams sent from the SUT that are meant to be synchronized, are reasonably possible to synchronize based on RTP and RTCP information and RTP packet jitter for those RTP streams. It cannot and does not test how closely they will be synchronized on playout in the receiver since the details of RTP stream synchronization depends on the receiver implementation.
The test result is agnostic to RTP packets being received by the SUT or not.

6.2.5 Source Description (SDES) Tests

6.2.5.1 Basic SDES Test

Purpose: Test if RTCP SDES packets sent from the SUT fulfils a few basic criteria for content and length consistency.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.2.6.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: Observe the SUT output.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least three RTCP packets are sent from the SUT per used SSRC identifier (at least the first RTCP packet may be sent before first RTP packet for each SSRC).

Pass criteria: The following is fulfilled for all RTCP SDES packets sent from the SUT:
1) The source count (SC) header field matches the number of SDES item chunks included in the respective RTCP SDES packet.
2) Each SDES item chunk ends with at least one octet containing a zero value.
3) Each SDES item in a chunk has a length field that corresponds to the text content length in octets of that SDES item, corresponding to the distance in octets starting from the octet after that length field until the start of the next SDES item or until the terminating zero value octet (see above).
4) Each SDES item in a chunk with non-zero content length has a text content where the last octet value is not zero (is not a zero-terminated string).

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.5.2 Canonical End-point Identifier (CNAME) Test

Purpose: Test if CNAME content is identical for every RTCP packet sent from the SUT belonging to the same RTP stream.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.5.1.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe the SUT output.
2) Note the "SSRC of sender" field (s_i) and CNAME SDES item field (c_i) values of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.

Stop condition: 1) RTP packets are sent from the SUT.
2) At least three RTCP packets are sent from the SUT per used SSRC identifier after sending the first RTP packet from the SUT for that SSRC identifier (one or more RTCP packets may be sent before first RTP packet for each SSRC).

Pass criteria: $c_i = c_j, for\ all\ i \in [1..n]\ and\ j \in [1..n]\ where\ s_i = s_j.$

Comments: The test result is agnostic to RTP packets being received by the SUT or not.

6.2.6 Receiver Report Block Tests

6.2.6.1 SSRC Consistency Test

Purpose: Test if the SSRC value included in a RTCP report block sent from the SUT corresponds to the SSRC of an RTP stream received by the SUT.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.2.6, and 6.2.2.7.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send RTP.

Test procedure: Observe the SUT and data injection output.

Stop condition: An RTCP packet is sent from the SUT, containing at least a receiver report block.

Pass criteria: The SSRC value in the RTCP report block sent from the SUT can be found as RTP header SSRC field value in at least one of the RTP packets sent from data injection.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.
A report block sent by the SUT containing an SSRC value identical to the SSRC value in the RTP header of RTP packets received by the SUT is henceforth denoted as the SUT "reporting on" the RTP stream those RTP packets belong to.

6.2.6.2 SSRC Completeness Test

Purpose: Test if all RTP streams (all SSRC) sent by data injection and received by the SUT in the RTP session are reported on by the SUT, while allowing that not all RTP streams are reported in every RTCP packet (e.g. due to RTCP packet MTU restrictions).

Status: Conditionally Mandatory (Precondition 1)

Preconditions: 1) The SUT is capable of using multiple simultaneous RTP streams in a single RTP session.
2) The SUT has passed test 6.2.6.1.
3) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.

4) Data injection is set to send RTP.
5) The number of simultaneously used, unique, SSRC values sent by the data injection in the RTP session (m) is known a-priori (only one in the single-stream case).

Test procedure: 1) Observe SUT and data injection output.
2) Note which SSRC values (S_i) that are included in RTCP report blocks sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Note which SSRC values (R_j) that are included in RTP headers of RTP packets sent from data injection.
5) Calculate the number of unique SSRC values (k) in report blocks sent from the SUT.

Stop condition: 1) At least one RTP packet with SSRC value R_j is sent from data injection for all $j \in [1..m]$.
2) $k = m$.

Pass criteria: $\{S_i\} = \{R_j\}\, for\ all\ i \in [1..n]\ and\ j \in [1..m]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.3 SSRC Change Test

Purpose: Test if an RTP stream received by the SUT that changes SSRC is correctly reported on by the SUT, keeping reporting separate per used SSRC value.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.6.4 and 6.2.6.6.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC, S_i) with monotonically increasing RTP sequence number (SN), and with the first RTP packet sent containing a non-zero RTP sequence number (SN), with a packet loss status that can be changed during the test session according to the test procedure without change in session signalling towards the SUT.

Test procedure: 1) Observe SUT and data injection output.
2) Send RTP packets for SSRC S_i from data injection to the SUT with a low but nonzero amount of RTP packet loss. The exact amount of RTP packet loss and loss pattern is not important and may be chosen freely by the test equipment.
3) Continue sending RTP packets for SSRC S_i from data injection to the SUT, without any RTP packet loss.
4) Note the fraction lost (f_i) and cumulative number of packets lost (c_i) for an RTCP report block sent from the SUT, reporting on the RTP stream with SSRC S_i sent from data injection. There may be multiple such RTCP reports sent from the SUT and the test may choose any of those.
5) Change the used SSRC to $S_{i+1} \neq S_i$ for RTP packets sent from data injection to the SUT (and stop sending any RTP packets with SSRC S_i), with other RTP header fields across the transition kept as if SSRC was not changed (keeping payload type, sequence number incremented by one, timestamp incremented similar to before, etc), still without any RTP packet loss.
6) Note the fraction lost (f_{i+1}) and cumulative number of packets lost (c_{i+1}) for an RTCP report block sent from the SUT, reporting on the RTP stream with SSRC S_{i+1} sent from data injection. There may be multiple such RTCP reports sent from the SUT and the test shall choose the first one sent after the RTP SSRC change.

Stop condition: An RTCP packet is sent from the SUT, containing at least a receiver report block reporting on the RTP stream with SSRC S_{i+1} from data injection.

Pass criteria: 1) $c_i \neq 0$ and $f_i \neq 0$.
2) $c_{i+1} = 0$ and $f_{i+1} = 0$.

Comments: The interarrival jitter field of the RTCP receiver report is, like cumulative number of packets lost and fraction lost fields in this test, also not kept across SSRC changes but restart from zero jitter. It is however deliberately not included as part of this test because it is hard to require the test equipment to achieve entirely predictable jitter or not introduce any jitter at all, to be able to construct a repeatable

jitter pass criteria.
The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.4 Initial Zero Loss Test

Purpose: Test if loss-free reception of RTP packets by the SUT at RTP session start is correctly reported by the SUT as no loss in RTCP, even if the initial RTP sequence number is non-zero.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.6.1.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) without loss, with monotonically increasing RTP sequence number (SN), and with the first RTP packet sent containing a non-zero RTP sequence number (SN).

Test procedure: Observe the SUT and data injection output.

Stop condition: 1) RTP packets are sent from data injection.
2) An RTCP packet is sent from the SUT, containing at least a receiver report block reporting on the RTP stream from data injection.

Pass criteria: The report block in the RTCP packet sent from the SUT reporting on the RTP stream from data injection fulfils:
1) The fraction lost report block field is zero.
2) The cumulative number of packets lost report block field is zero.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.5 Zero Loss Test

Purpose: Test if loss-free reception of RTP packets by the SUT during a reporting period is correctly reflected in the RTCP report sent by the SUT.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.6.4.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) without loss and with monotonically increasing RTP sequence number (SN).

Test procedure: Observe the SUT and data injection output.

Stop condition: 1) RTP packets are sent from data injection.
2) Two RTCP packets are sent from the SUT, each containing at least a receiver report block reporting on the RTP stream from data injection.

Pass criteria: The report block in the RTCP packets sent from the SUT, reporting on the RTP stream from data injection fulfils:
1) The fraction lost report block field is zero in the second RTCP packet sent from the SUT during the test.
2) The cumulative number of packets lost report block field value is identical in the first and second RTCP packet sent from the SUT during the test.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.6 Loss Test

Purpose: Test if the "fraction lost" and "cumulative number of packets lost" fields in the receiver report sent from the SUT are consistent with one another and with the "extended last sequence number received" field in the same receiver report.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.6.5 and 6.2.6.11.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) with loss according to test procedure (below) and with monotonically increasing RTP sequence number (SN).

Test procedure: 1) Observe the SUT and data injection output.
2) Note the extended last sequence number received (s_i), fraction lost (f_i) and cumulative number of packets lost (c_i) for each RTCP report block sent from the SUT reporting on the RTP stream sent from data injection.
3) The below steps of this test procedure are designed to be run multiple times, using a few different loss patterns described in steps 5a)-5e) . Each repetition of the below steps may be run in immediate succession (successive RTCP reporting periods) or may be separated by loss-free RTCP periods. In either case, it is the responsibility of the test setup to ensure that the RTP packets dropped by data injection are reported on in the next RTCP report sent from the SUT, e.g. by letting data injection drop RTP packets just after an RTCP packet is sent from the SUT.
4) An RTCP packet, $i-1$, is sent from the SUT, reporting on the RTP stream from data injection. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure and i will be different across the repetitions related to steps 5a)-5e).
5) Drop RTP packets from data injection to the SUT according to the below sub-bullets, one sub-bullet per repetition, k, of steps 4-10, where n_k is the number of dropped RTP packets in the RTCP reporting period for that repetition:

a) A single RTP packet; $k=1, n_1=1$.
b) Two RTP packets in succession; $k=2, n_2=2$.
c) Two RTP packets with three RTP packets in between; $k=3, n_3=2$.
d) Every 20th RTP packet; $k=4, n_4={N_k}/{20}$.
e) Every 10th RTP packet; $k=5, n_5={N_k}/{10}$.

6) An RTCP packet, i, is sent from the SUT, reporting on the RTP stream from data injection.
7) Calculate number of packets sent (disregarding packet drops) $N_k=s_i-s_{i-1}$.
8) Calculate fraction lost $F_k=\left\lfloor 256\,{n_k}/{N_k}\right\rfloor$, with n_k set according to 5a) – 5e) above.
9) Calculate packets lost $L_k=c_i-c_{i-1}$.
10) Repeat from step 4.

Stop condition: Step 9 is fulfilled for $k=5$, repetition 5e).

Pass criteria: 1) $f_i=F_k, for\ all\ k\in[1..5]$.
2) $L_k=\lfloor n_k\rfloor, for\ all\ k\in[1..5]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.
It is assumed that the information in extended last sequence number received is correct (see test 6.2.6.11) and enough to correlate with actual, observed RTP packet loss.

6.2.6.7 Wrapped Loss Test

Purpose: Test if the "fraction lost" and "cumulative number of packets lost" fields in the receiver report sent from the SUT are consistent with one another and with the "extended last sequence number received" field in the same receiver report, even when the RTP "sequence number" field in RTP packets sent from data injection wraps around.

Status: Conditionally Mandatory (Precondition 4)

Preconditions: 1) The SUT has passed tests 6.2.6.6 and 6.2.6.12.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) with loss according to test procedure (below).
4) The "sequence number" (SN) field in the first RTP packet sent from data injection is set to a value

close to 2^{16}-1 (when interpreted as an unsigned 16-bit integer) in step 5 of the test procedure, such that the value will wrap during each repetition of the test.

Test procedure: 1) Observe the SUT and data injection output.
2) Note the extended last sequence number received (s_i), fraction lost (f_i) and cumulative number of packets lost (c_i) for each RTCP report block sent from the SUT reporting on the RTP stream sent from data injection.
3) The below steps of this test procedure are designed to be run multiple times, using a few different loss patterns described in steps 5a)-5e) . Each repetition of the below steps may be run in immediate succession (successive RTCP reporting periods) or may be separated by loss-free RTCP periods. In either case, it is the responsibility of the test setup to ensure that the RTP packets dropped by data injection are reported on in the next RTCP report sent from the SUT, e.g. by letting data injection drop RTP packets just after an RTCP packet is sent from the SUT.
4) An RTCP packet, $i-1$, is sent from the SUT, reporting on the RTP stream from data injection. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure and i will be different across the repetitions related to steps 5a)-5e).
5) Drop RTP packets from data injection to the SUT according to the below sub-bullets, one sub-bullet per repetition, k, of steps 4-10, where n_k is the number of dropped RTP packets in the RTCP reporting period for that repetition:
a) A single RTP packet; $k=1, n_1=1$.
b) Two RTP packets in succession; $k=2, n_2=2$.
c) Two RTP packets with three RTP packets in between; $k=3, n_3=2$.
d) Every 20th RTP packet; $k=4, n_4={}^{N_k}/_{20}$.
e) Every 10th RTP packet; $k=5, n_5={}^{N_k}/_{10}$.
6) An RTCP packet, i, is sent from the SUT, reporting on the RTP stream from data injection.
7) Calculate number of packets sent (disregarding packet drops) $N_k=s_i-s_{i-1}$.
8) Calculate fraction lost $F_k=\left\lfloor 256\,{}^{n_k}/_{N_k}\right\rfloor$, with n_k set according to 5a) – 5e) above.
9) Calculate packets lost $L_k=c_i-c_{i-1}$.
10) Repeat from step 4.

Stop condition: Step 9 is fulfilled for $k=5$, repetition 5e).

Pass criteria: 1) $f_i=F_k, for\ all\ k\in[1..5]$.
2) $L_k=\lfloor n_k\rfloor, for\ all\ k\in[1..5]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.
It is assumed that the information in extended last sequence number received is correct (see test 6.2.6.12) and enough to correlate with actual, observed RTP packet loss.

6.2.6.8 Duplicate Loss Test

Purpose: Test if the "fraction lost" and "cumulative number of packets lost" fields in the receiver report sent from the SUT are consistent with one another and with the "extended last sequence number received" field in the same receiver report, even when receiving duplicate RTP packets (with duplicate values in "sequence number" field) sent from data injection.

Status: Conditionally Mandatory (Precondition 4)

Preconditions: 1) The SUT has passed test 6.2.6.6.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) with loss according to test procedure (below).
4) Data injection is set to send some RTP packets multiple times according to test procedure (below).

Test procedure: 1) Observe the SUT and data injection output.
2) Note the extended last sequence number received (s_i), fraction lost (f_i) and cumulative number of packets lost (c_i) for each RTCP report block sent from the SUT reporting on the RTP stream sent from data injection.

3) The below steps of this test procedure are designed to be run multiple times, using a few different loss and duplication patterns described in steps 5a)-5e) . Each repetition of the below steps may be run in immediate succession (successive RTCP reporting periods) or may be separated by loss-free RTCP periods. In either case, it is the responsibility of the test setup to ensure that the RTP packets dropped or duplicated by data injection are reported on in the next RTCP report sent from the SUT, e.g. by letting data injection drop and duplicate RTP packets just after an RTCP packet is sent from the SUT.
4) An RTCP packet, $i-1$, is sent from the SUT, reporting on the RTP stream from data injection. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure and i will be different across the repetitions related to steps 5a)-5e).
5) Drop and duplicate RTP packets from data injection to the SUT according to the below sub-bullets, one sub-bullet per repetition, k, of step 4-10, where n_k is the number of dropped RTP packets and r_k is the number of duplicated RTP packets in the RTCP reporting period for that repetition:

a) Drop a single RTP packet, duplicate a single non-dropped RTP packet;
$k = 1, n_1 = 1, r_1 = 1.$

b) Drop two RTP packets in succession, duplicate a single non-dropped RTP packet;
$k = 2, n_2 = 2, r_2 = 1.$

c) Drop two RTP packets with three RTP packets in between, duplicate three non-dropped RTP packets;
$k = 3, n_3 = 2, r_3 = 3.$

d) Drop every 20th RTP packet, duplicate every 20th non-dropped RTP packet;
$k = 4, n_4 = {N_k}/{20}, r_4 = {19N_k}/{400}.$

e) Drop every 10th RTP packet, duplicate every 20th non-dropped RTP packet;
$k = 5, n_5 = {N_k}/{10}, r_4 = {9N_k}/{200}.$

6) An RTCP packet, i, is sent from the SUT, reporting on the RTP stream from data injection.
7) Calculate number of packets sent (disregarding packet drops and duplications) $N_k = s_i - s_{i-1}$.
8) Calculate fraction lost $F_k = \left\lfloor 256\, {(n_k - r_k)}/{N_k} \right\rfloor$, with n_k and r_k set according to 5a) – 5e) above.
9) Calculate packets lost $L_k = c_i - c_{i-1}$.
10) Repeat from step 4.

Stop condition: Step 9 is fulfilled for $k = 5$, repetition 5e).

Pass criteria: 1) $f_i = F_k, for\ all\ k \in [1..5]$.
2) $L_k = \lfloor n_k - r_k \rfloor, for\ all\ k \in [1..5]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.
It is assumed that the information in extended last sequence number received is correct (see test 6.2.6.11) and enough to correlate with actual, observed RTP packet loss.

6.2.6.9 Out-of-sequence Loss Test

Purpose: Test if the "fraction lost" and "cumulative number of packets lost" fields in the receiver report sent from the SUT are consistent with one another and with the "extended last sequence number received" field in the same receiver report, even when the SUT is receiving RTP packets with the "sequence number" field not monotonically incrementing.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed test 6.2.6.6.
2) The SUT set to receive RTP packets during the test session, e.g. is pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) with loss according to test procedure (below) and where the "sequence number" field in the RTP packet header is not generated with a constant increment of 1 (n, n+1, n+2, …) but with alternating increments of -1 and 3 (n+1, n, n+3, n+2, …).

Test procedure: 1) Observe the SUT and data injection output.
2) Note the extended last sequence number received (s_i), fraction lost (f_i) and cumulative number of packets lost (c_i) for each RTCP report block sent from the SUT reporting on the RTP stream sent

from data injection.
3) The below steps of this test procedure are designed to be run multiple times, using a few different loss patterns described in steps 5a)-5e) . Each repetition of the below steps may be run in immediate succession (successive RTCP reporting periods) or may be separated by loss-free RTCP periods. In either case, it is the responsibility of the test setup to ensure that the RTP packets dropped by data injection are reported on in the next RTCP report sent from the SUT, e.g. by letting data injection drop RTP packets just after an RTCP packet is sent from the SUT.
4) An RTCP packet, $i-1$, is sent from the SUT, reporting on the RTP stream from data injection. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure and i will be different across the repetitions related to steps 5a)-5e).
5) Drop RTP packets from data injection to the SUT according to the below sub-bullets, one sub-bullet per repetition, k, of steps 4-10, where n_k is the number of dropped RTP packets in the RTCP reporting period for that repetition:

a) A single RTP packet;
$k=1, n_1=1$.
b) Drop two RTP packets in succession;
$k=2, n_2=2$.
c) Drop two RTP packets with three RTP packets in between;
$k=3, n_3=2$.
d) Drop every 20th RTP packet;
$k=4, n_4={}^{N_k}/_{20}$.
e) Drop every 10th RTP packet;
$k=5, n_5={}^{N_k}/_{10}$.

6) An RTCP packet, i, is sent from the SUT, reporting on the RTP stream from data injection.
7) Calculate number of packets sent (disregarding packet drops) $N_k=s_i-s_{i-1}$.
8) Calculate fraction lost $F_k=\left\lfloor 256\,{}^{n_k}/_{N_k}\right\rfloor$, with n_k set according to 5a) – 5e) above.
9) Calculate packets lost $L_k=c_i-c_{i-1}$.
10) Repeat from step 4.

Stop condition: Step 9 is fulfilled for$k=5$, repetition 5e).

Pass criteria: 1) $f_i=F_k, for\ all\ k\in[1..5]$.
2) $L_k=\lfloor n_k\rfloor, for\ all\ k\in[1..5]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.
It is assumed that the information in extended last sequence number received is correct (see test 6.2.6.11) and enough to correlate with actual, observed RTP packet loss.

6.2.6.10 All Loss Test

Purpose: Test if no RTP packets being received by the SUT during one or more reporting periods is correctly reflected in the RTCP reports sent by the SUT when receiving RTP packets again.

Status: Mandatory

Preconditions: 1) The SUT has passed tests 6.2.6.4 and 6.2.6.5.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) with loss according to the test procedure below.

Test procedure: 1) Observe SUT and data injection output.
2) Note the cumulative number of packets lost (c_i) for each RTCP report block sent from the SUT reporting on the RTP stream sent from data injection.
3) RTP packets are sent from data injection.
4) An RTCP packet, $i=1$, is sent from the SUT, reporting on the RTP stream from data injection. This RTCP packet is not necessarily the very first RTCP packet sent by the SUT in the RTP session but just the first in the scope of this test. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure.

5) All RTP packets are prevented from reaching the SUT and dropped (not buffered), without modifying the session as seen from the SUT, e.g. through proprietary data injection interaction or through transport network actions.
6) Note the sequence number s_1 of the last RTP packet reaching the SUT before the RTP loss period.
6) At least two RTCP packets, $i \in [2..n]$, are sent from the SUT, reporting on the RTP stream from data injection (the first one likely covering the beginning of the all-loss period).
7) RTP packets are no longer prevented from reaching the SUT, without modifying the session as seen from the SUT, e.g. through proprietary data injection interaction or through transport network actions.
8) Note the sequence number s_2 of the first RTP packet reaching the SUT after the RTP loss period.

Stop condition: An RTCP packet, $i = n + 1$, is sent from the SUT after step 8 in the test procedure.

Pass criteria: The report block in RTCP packets sent from the SUT, reporting on the RTP stream from data injection fulfils:
1) RTCP packets with $i \in [3..n]$ do not contain any report block for the SSRC used by data injection and that was subjected to loss in step 5 of the test procedure.
2) $c_{n+1} = c_1 + (s_2 - s_1)$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not. s_2 must be strictly larger than s_1 (i.e., not wrapped around) for the pass criteria to work as currently formulated.

6.2.6.11 Extended Highest Sequence Number Received Test

Purpose: Test if the "extended last sequence number received" field in the receiver report sent from the SUT is consistent with RTP "sequence number" field sent by data injection.

Status: Mandatory

Preconditions: 1) The SUT has passed test 6.2.6.1.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC).
4) The "sequence number" (SN) field in the first RTP packet sent from data injection is monotonously increasing (see 6.2.6.12 for test of wrapped value).

Test procedure: 1) Observe the SUT and data injection output.
2) Note the "extended last sequence number received" field (e_i) of each RTCP packet sent from the SUT .
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count the number of sent RTP packets (P_i) sent from data injection between each pair $\{i - 1, i\}$ of subsequent RTCP packets sent from the SUT. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure.
5) Note the "sequence number" field ($s_{i,j}$) of each RTP packet sent by data injection, with $j \in [1..P_i]$.
6) Note the RTP packet number, $l_i = j$, where $s_{i,j} = (e_i \,\%\, 2^{16})$, or if no RTP packet with such $s_{i,j}$ exists, set $l_i = 0$.

Stop condition: 1) RTP packets are sent from data injection.
2) At least three RTCP packets are sent from the SUT after the first RTP packet from data injection (one or more RTCP packets may be sent before first RTP packet).

Pass criteria: $l_i \in [1..P_i], for\ all\ i \in [2..n]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.12 Wrapped Extended Highest Sequence Number Received Test

Purpose: Test if the "extended last sequence number received" field in the receiver report sent from the SUT is consistent with RTP "sequence number" field sent by data injection, even when the "sequence number" field wraps around, and the extended part above 16 bits sent from the SUT is properly incremented after wrap.

Status: Conditionally Mandatory (Precondition 4)

Preconditions: 1) The SUT has passed test 6.2.6.11.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC).
4) The "sequence number" (SN) field in the first RTP packet sent from data injection is set to a value close to 2^{16}-1 (when interpreted as an unsigned 16-bit integer), such that the value will wrap during the test.

Test procedure: 1) Observe the SUT and data injection output.
2) Note the "extended last sequence number received" field (e_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count the number of sent RTP packets (P_i) sent from data injection between each pair $\{i-1,i\}$ of subsequent RTCP packets sent from the SUT. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure.
5) Note the "sequence number" field ($s_{i,j}$) of each RTP packet send by data injection, with $j \in [1..P_i]$.
6) Note the RTP packet number, $l_i = j$, where $s_{i,j} = (e_i \% 2^{16})$, or if no RTP packet with such $s_{i,j}$ is received, set $l_i = 0$.

Stop condition: 1) RTP packets are sent from data injection.
2) At least three RTCP packets are sent from the SUT after the first RTP packet from data injection (one or more RTCP packets may be sent before first RTP packet).
3) $s_{1,1} > s_{n-1,P_{n-1}}$ when interpreted as unsigned 16-bit integers.

Pass criteria: 1) $l_i \in [1..P_i], for\ all\ i \in [2..n]$.
2) $(e_n \gg 16) - (e_1 \gg 16) = 1$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.13 Out-of-sequence Extended Highest Sequence Number Received Test

Purpose: Test if the "extended last sequence number received" field in the receiver report sent from the SUT is consistent with RTP "sequence number" field sent by data injection, even when the "sequence number" field in RTP packets sent by data injection is not monotonically incrementing.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed test 6.2.6.11.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) where the "sequence number" field in the RTP packet header is not generated with a constant increment of 1 (n, n+1, n+2, …) but with alternating increments of -1 and 3 (n+1, n, n+3, n+2, …).

Test procedure: 1) Observe the SUT and data injection output.
2) Note the "extended last sequence number received" field (e_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Count the number of sent RTP packets (P_i) sent from data injection between each pair $\{i-1,i\}$ of subsequent RTCP packets sent from the SUT. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure.
5) Note the "sequence number" field ($s_{i,j}$) of each RTP packet sent by data injection, with $j \in [1..P_i]$.
6) Note the RTP packet number, $l_i = j$, where $s_{i,j} = (e_i \% 2^{16})$, or if no RTP packet with such $s_{i,j}$ exists, set $l_i = 0$.

Stop condition: 1) RTP packets are sent from data injection.
2) At least three RTCP packets are sent from the SUT after the first RTP packet from data injection (one or more RTCP packets may be sent before first RTP packet).

Pass criteria: 1) $l_i \in [1..P_i], for\ all\ i \in [2..n]$.
2) $(e_i \% 2^{16}) = \max_j s_{i,j}, for\ all\ i \in [2..n]\ and\ j \in [1..l_i]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.14 Interarrival Jitter Test

Purpose: Test if the "interarrival jitter" field in the receiver report sent from the SUT, as described by RTP [2] section 6.4.1, is consistent with RTP packets sent by data injection.

Status: Conditionally Mandatory (Precondition 3)

Preconditions: 1) The SUT has passed tests 6.2.4.2, 6.2.4.4, and 6.2.6.1.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC) that is subjected to delay variations according to delay and loss profiles from TS 26.132 [11] Annex E.3.

Test procedure: 1) Observe the SUT and data injection output.
2) Retrieve RTP Time Stamp rate information (S), e.g. from "<clock rate>" on the SDP "a=rtpmap:<payload type> <encoding name>/<clock rate>" line used to set up the SUT session, applicable for the RTP stream that the RTCP packets sent from the SUT are associated with.
3) For each delay and loss profile k from dly_profile_20msDRX_10pctBLER_e2e and dly_profile_20msDRX_10pctBLER_ue1_to_eNB2 in TS 26.131 [11] Annex E.3 (see also Table 6.2.6.14-1 below), repeat the following steps:
4) Start data injection sending RTP packets subjected to delay variations according to the delay and loss profile.
5) Note the reception time ($t_{k,i}$) and "interarrival jitter" ($j_{k,i}$) field in the report block of each RTCP packet sent from the SUT.
6) Count the number of RTCP packets (n_k) sent from the SUT.
7) Count the number of sent RTP packets ($P_{k,i}$) sent from data injection between each pair $\{i-1, i\}$ of subsequent RTCP packets sent from the SUT. The value of i and its relation to how many RTCP packets that are actually sent by the SUT is only significant in context of describing the test procedure.
8) Note the reception time ($r_{k,i,j}$) and "timestamp" field ($T_{k,i,j}$) of each RTP packet sent from data injection, with $j \in [1..P_{k,i}]$.
9) Calculate the arrival time $a_{k,i,j} = T_{k,1,1} + {(r_{k,i,j} - r_{k,1,1})}/{S}$ in timestamp units of each RTP packet sent from data injection, with $j \in [1..P_{k,i}]$, and where $r_{k,1,1}$ is the reception time of the first received RTP packet with timestamp $T_{k,1,1}$ sent from data injection.
10) Calculate the difference in RTP packet spacing $D_{k,i,j} = (a_{k,i,j} - T_{k,i,j}) - (a_{k,i,j-1} - T_{k,i,j-1})$, with $j \in [2..P_{k,i}]$.
11) Calculate (not using integer arithmetic) the filtered RTP packet jitter $J_{k,i} = \sum_{j=2}^{P_{k,i}} s_j$, where $s_j = {(|D_{k,i,j}| - s_{j-1})}/{16}$, with $s_1 = J_{k,i-1}$, where $J_{k,i-1}$ corresponds to the filtered RTP packet jitter reported in previous RTCP packet sent from the SUT, and $J_{k,1} = 0$.
12) When there are no more delay and loss profile data to apply to RTP packets, stop sending RTP packets to avoid further impact to $J_{k,i}$.

Stop condition: 1) RTP packets are sent from data injection.
2) At least one RTCP packet is sent from the SUT after the last RTP packet from data injection, for each delay and loss profile.

Pass criteria: $j_{k,n_k} = \lfloor J_{k,n_k} \rfloor$, for all k, where expected, nominal values based only on delay and loss profile content, excluding any data injection or test instrument implementation impact, are (for information):

Table 6.2.6.14-1 Examples of delay and loss profile jitter

Delay and loss profile	k	n_k	$\lfloor J_{k,n_k} \rfloor$ $S = 8000$	$\lfloor J_{k,n_k} \rfloor$ $S = 16000$
dly_profile_20msDRX_10pctBLER_e2e	1	8000	93	186
dly_profile_20msDRX_10pctBLER_ue1_to_eNB2	2	8000	27	55

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.15 Last Sender Report Timestamp Test

Purpose: Test if the LSR field in the receiver report from the SUT corresponds to middle 32 bits of "NTP timestamp" 64-bit field for some previous RTCP sender report sender info sent from data injection.

Status: Mandatory

Preconditions:
1) The SUT has passed test 6.2.2.6.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC).

Test procedure:
1) Observe the SUT and data injection output.
2) Note the LSR field (L_i) in the report block of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Note the "NTP timestamp, most significant word" (m_j) and "NTP timestamp, least significant word" (l_j) fields in the sender information of each RTCP packet sent from data injection.
5) Count the number of RTCP packets (p) sent from data injection.
6) Calculate the reduced-precision last SR NTP time $N_j = \left(\left(m_j \,\&\, (2^{16}-1)\right) \ll 2^{16}\right) + \left(l_j \gg 2^{16}\right)$ for each RTCP packet sent from data injection, with $j \in [1..p]$.
7) Note the RTCP packet number sent from data injection, $q_i = j$, where $L_i = N_j$, or if no RTCP packet with such N_j exists, set $q_i = 0$.

Stop condition:
1) RTP packets are sent from data injection.
2) RTCP packets are sent from data injection.
3) At least three RTCP packets with $L_i \neq 0$ are sent from the SUT after the first RTCP packet is sent from data injection.

Pass criteria: $q_i > 0, for\ at\ least\ one\ i \in [1..n]$.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.6.16 Delay Since Last SR Test

Purpose: Test if the reception time and delay since last SR field in the receiver report from the SUT is consistent with observed reception time and NTP timestamp field of the corresponding RTCP sender report sender info from data injection.

Status: Mandatory

Preconditions:
1) The SUT has passed test 6.2.6.15.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to send a single RTP stream (one SSRC).

Test procedure:
1) Observe the SUT and data injection output.
2) Note the LSR (L_i) and "delay since last LSR" (DLSR) (D_i) fields in the report block and reception time (t_i) of each RTCP packet sent from the SUT.
3) Count the number of RTCP packets (n) sent from the SUT.
4) Note the "NTP timestamp, most significant word" (m_j) and "NTP timestamp, least significant word" (l_j) fields in the sender information, and reception time (r_j) of each RTCP packet sent from data injection.
5) Count the number of RTCP packets (p) sent from data injection.
6) Calculate the reduced-precision last SR NTP time $N_j = \left(\left(m_j \,\&\, (2^{16}-1)\right) \ll 2^{16}\right) + \left(l_j \gg 2^{16}\right)$ for each RTCP packet sent from data injection, with $j \in [1..p]$.
7) Calculate the data injection reporting interval $s_j = r_j - r_{j-1}$, for $j \in [2..p]$.
8) Note the RTCP packet number sent from data injection, $q_i = j$, where $L_i = N_j$, or if no RTCP packet with such N_j exists, set $q_i = 0$.

9) Calculate the observed round-trip reporting time, $d_i = r_{q_i} - t_i$ for all i with $q_i \neq 0$, where r_{q_i} is the r_j with $q_i = j$.

10) Calculate the round-trip time, $R_i = r_{q_i} - t_i - {}^{D_i}/_{65536}$ for all i with $q_i \neq 0$.

Stop condition: 1) RTP packets are sent from data injection.
2) RTCP packets are sent from data injection.
3) At least three RTCP packets with $L_i \neq 0$ and $D_i \neq 0$ are sent from the SUT after the first RTCP packet from data injection.

Pass criteria: 1) $d_i \geq {}^{D_i}/_{65536}$ for all i with $q_i \neq 0$, i.e. the DLSR value cannot represent a longer time than the entire, observed round-trip time.
2) ${}^{D_i}/_{65536} \leq \max_j s_j$ for all i with $q_i \neq 0$ and $j \in [2..p]$, i.e. the DLSR value cannot represent a time that exceeds an RTCP reporting interval.

Comments: This test only performs a sanity check of the DLSR field sent from the SUT since the DLSR field value in general cannot be verified without close knowledge of SUT detailed conditions and implementation.
The test result is agnostic to RTP packets being sent by the SUT or not.

6.2.7 Feedback Report Block Tests

6.2.7.1 Ignoring Unknown Feedback Report Test

Purpose: Test if receiving an RTCP feedback report [7] of unknown (non-negotiated) type is correctly ignored by the SUT without negative impact on other RTP/RTCP operation.

Status: Mandatory

Preconditions: 1) The SUT is set to not use RTCP Feedback messages during the test session, e.g. pre-configured or through no "a=rtcp-fb" lines being included in SDP signalling from the test instrument.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe SUT output.
2) RTP and RTCP packets are sent from the SUT.
3) Data injection sends an RTCP Feedback message to the SUT, as part of a compound RTCP message starting with a RTCP SR or RR. The RTCP Feedback message is characterized by the Payload Type (PT) field in the RTCP header set to either 205 (RTPFB) or 206 (PSFB), and the Feedback Message Type (FMT) may be chosen freely by data injection and set to any value.

Stop condition: One or more RTP packets and one or more RTCP packets are sent by the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: RTP and RTCP packets are sent by the SUT after step 3 in the test procedure.

Comments: The test only tries to detect if the RTP/RTCP stack in the SUT is operational after receiving an unexpected RTCP FB message in the sense that RTP and RTCP packets are still sent from the SUT. The test does not check for more detailed signs of SUT RTP/RTCP stack malfunction after receiving the unexpected RTCP message.
The test result is agnostic to RTP packets being received by the SUT or not.

6.2.8 Extended Report Block Tests

6.2.8.1 Ignoring Unknown XR Test

Purpose: Test if receiving an RTCP XR [13] packet of unknown (non-negotiated) type is correctly ignored by the SUT without negative impact on other RTP/RTCP operation.

Status: Mandatory

Preconditions: 1) The SUT is set to not use RTCP XR messages during the test session, e.g. pre-configured or through no "a=rtcp-xr" lines being included in SDP signalling from the test instrument.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe SUT output.
2) RTP and RTCP packets are sent from the SUT.
3) Data injection sends an RTCP Feedback message to the SUT, as part of a compound RTCP message starting with a RTCP SR or RR. The RTCP Feedback message is characterized by the Payload Type (PT) field in the RTCP header set to 207, and the reserved bits are all set to zero.

Stop condition: One or more RTP packets and one or more RTCP packets are sent by the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: RTP and RTCP packets are sent by the SUT after step 3 in the test procedure.

Comments: The test only tries to detect if the RTP/RTCP stack in the SUT is operational after receiving an unexpected RTCP XR message in the sense that RTP and RTCP packets are still sent from the SUT. The test does not check for more detailed signs of SUT RTP/RTCP stack malfunction after receiving the unexpected RTCP message.
The test result is agnostic to RTP packets being received by the SUT or not.

6.2.9 APP Tests

6.2.9.1 Ignoring Unknown APP Test

Purpose: Test if receiving an RTCP APP packet (see section 6.7 of IETF RFC 3550 [2]) with unknown name is correctly ignored by the SUT without negative impact on other RTP/RTCP operation.

Status: Mandatory

Preconditions: 1) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe SUT output.
2) RTP and RTCP packets are sent from the SUT.
3) Data injection sends an RTCP APP message to the SUT, as part of a compound RTCP message starting with a RTCP SR or RR. The RTCP APP message is characterized by the Payload Type (PT) field in the RTCP header set to 204, the subtype field may be chosen freely by data injection and set to any value, and the name field may be chosen freely by data injection from any sequence of four ASCII characters, with the exception that if the SUT is known to explicitly recognize APP messages with certain name field values, the name field is not set to any of those values.

Stop condition: One or more RTP packets and one or more RTCP packets are sent by the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: RTP and RTCP packets are sent by the SUT after step 3 in the test procedure.

Comments: The test only tries to detect if the RTP/RTCP stack in the SUT is operational after receiving an unexpected RTCP APP message in the sense that RTP and RTCP packets are still sent from the SUT. The test does not check for more detailed signs of SUT RTP/RTCP stack malfunction after receiving the unexpected RTCP message.
The test result is agnostic to RTP packets being received by the SUT or not.

6.2.10 Reduced-Size Packet Tests

6.2.10.1 Ignoring Unsupported Reduced-Size Test

Purpose: Test if receiving a reduced-size RTCP packet [8] when its use is not negotiated is correctly ignored by the SUT without negative impact on other RTP/RTCP operation.

Status: Mandatory

Preconditions: 1) The SUT is set to not use reduced-size RTCP messages during the test session, e.g. pre-configured or through no "a=rtcp-rsize" lines being included in SDP signalling from the test instrument.
2) The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe SUT output.
2) RTP and RTCP packets are sent from the SUT.
3) Data injection sends one or more compound RTCP messages to the SUT.
4) Data injection sends a reduced-size RTCP message to the SUT, not a compound RTCP starting with a RTCP SR or RR. What reduced-size RTCP message to use may be chosen freely by data injection as long as it is a correct and well-formed RTCP message, but can e.g. be a single RTCP Generic NACK (see section 6.2.1 of IETF RFC 4585 [7]).

Stop condition: One or more RTP packets and one or more RTCP packets are sent by the SUT, or time has passed corresponding to at least three maximum RTCP intervals (see clause 6.2.1).

Pass criteria: RTP and RTCP packets are sent by the SUT after step 4 in the test procedure.

Comments: The test only tries to detect if the RTP/RTCP stack in the SUT is operational after receiving an unexpected reduced-size RTCP message in the sense that RTP and RTCP packets are still sent from the SUT. The test does not check for more detailed signs of SUT RTP/RTCP stack malfunction after receiving the unexpected RTCP message.
The test result is agnostic to RTP packets being received by the SUT or not.

6.3 RTP Tests

6.3.1 General

This clause describes tests that are applicable to all RTP and SRTP packets. If not explicitly stated otherwise, any reference to RTP is equally applicable to SRTP. Tests that are only applicable to SRTP and not to RTP are specified in clause 6.5.

6.3.2 Basic RTP Tests

6.3.2.1 Receive RTP Padding Test

Purpose: Test if RTP packets with padding are correctly received by the SUT.

Status: Mandatory

Preconditions: 1) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
2) Data injection is set to use RTP padding as described by sections 4 and 5.1 of RFC 3550 in some or all RTP packets during the test session (RTP header P bit set). Any number of padding octets (larger than zero) suitable to the test instrument may be chosen.

Test procedure: 1) Observe the SUT application-level output based on RTP packet payload, e.g. audio from a speaker and/or video on a screen.
2) RTP packets with RTP payload and RTP padding are sent from data injection.

Stop condition: Ten or more seconds have passed after first RTP packet was sent from data injection.

Pass criteria: Application-level output from the SUT exist as expected (e.g. audio and/or video) and is undistorted compared to a corresponding test setup where RTP padding is not used.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.3.2.2 Initial SSRC Value Test

Purpose: Test if the SSRC value is chosen randomly by the SUT for every new RTP stream sent by the SUT.

Status: Mandatory

Preconditions: The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe the SUT output.
2) Note synchronization source (SSRC) field values (s_i) in RTP packet headers from different RTP streams (i) sent from the SUT.
3) Start sending three or more new RTP streams from the SUT, one after the other (one at a time) or sent simultaneously. Any method suitable to the SUT and the test instrument may be chosen to achieve this, e.g. making three or more calls with the SUT, adding and removing an RTP stream three or more times during a single session (call), or adding three or more simultaneous RTP streams for a single session (call).
4) Count the number of different RTP streams (different SSRC values), n, sent during the test from the SUT.

Stop condition: RTP packets for three or more, separate RTP streams are sent from the SUT.

Pass criteria: 1) $s_i \neq s_{i-1} \; for \; all \; i \in [2..n]$.
2) $s_i \neq s_{i-1} + 1 \; for \; all \; i \in [2..n]$.
3) $s_i \neq s_{i-1} - 1 \; for \; all \; i \in [2..n]$.

Comments: The test pass criteria are a rough approximation of testing that the SSRC is a random value, i.e. simply testing that SSRC values are not all the same and not an increasing or decreasing sequence. Testing that the SSRC value is truly a random number would require a much more elaborate formula. The test result is agnostic to RTP packets being received by the SUT or not.

6.3.2.3 Initial Sequence Number Test

Purpose: Test if the Sequence Number start value is chosen randomly by the SUT for every new RTP stream sent by the SUT.

Status: Mandatory

Preconditions: The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe the SUT output.
2) Note the sequence number (SN) field value (S_i) of the first RTP packet header of each RTP stream (i) sent from the SUT.
3) Start sending three or more new RTP streams from the SUT, one after the other (one at a time) or sent simultaneously. Any method suitable to the SUT and the test instrument may be chosen to achieve this, e.g. making three or more calls with the SUT, adding and removing an RTP stream three or more times during a single session (call), or adding three or more simultaneous RTP streams for a single session (call).
4) Count the number of different RTP streams (different SSRC values), n, sent during the test from the SUT.

Stop condition: RTP packets for three or more, separate RTP streams are sent from the SUT.

Pass criteria: 1) $S_i \neq S_{i-1} \; for \; all \; i \in [2..n]$.
2) $S_i \neq S_{i-1} + 1 \; for \; all \; i \in [2..n]$.
3) $S_i \neq S_{i-1} - 1 \; for \; all \; i \in [2..n]$.

Comments: The test pass criteria are a rough approximation of testing that the first SN is a random value, i.e. simply testing that the first SN values are not all the same and not an increasing or decreasing sequence. Testing that the first SN value is truly a random number would require a much more elaborate formula. The test result is agnostic to RTP packets being received by the SUT or not.

6.3.2.4 Initial Time Stamp Test

Purpose: Test if the timestamp start value is chosen randomly by the SUT for every new RTP stream sent by the SUT.

Status: Mandatory

Preconditions: The SUT is set to send RTP packets during the test session, e.g. pre-configured or set as send-receive or send-only through test instrument signalling.

Test procedure: 1) Observe the SUT output.
2) Note the timestamp field value (T_i) of the first RTP packet header of each RTP stream (i) sent from the SUT.
3) Start sending three or more new RTP streams from the SUT, one after the other (one at a time) or sent simultaneously. Any method suitable to the SUT and the test instrument may be chosen to achieve this, e.g. making three or more calls with the SUT, adding and removing a single RTP stream three or more times during a single session (call), or adding three or more simultaneous RTP streams for a single session (call).
4) Count the number of different RTP streams (different SSRC values), n, sent during the test from the SUT.

Stop condition: RTP packets for three or more, separate RTP streams have been sent from the SUT.

Pass criteria: 1) $T_i \neq T_{i-1}\ for\ all\ i \in [2..n]$.
2) $T_i \neq T_{i-1} + 1\ for\ all\ i \in [2..n]$.
3) $T_i \neq T_{i-1} - 1\ for\ all\ i \in [2..n]$.

Comments: The test pass criteria are a rough approximation of testing that the first TS is a random value, i.e. simply testing that the first TS values are not all the same and not an increasing or decreasing sequence. Testing that the first TS value is truly a random number would require a much more elaborate formula. The test result is agnostic to RTP packets being received by the SUT or not.

6.3.3 RTP Header Extension Tests

6.3.3.1 Ignore Unknown Header Extension Test

Purpose: Test if the SUT correctly ignores the RTP header extension when receiving RTP packets with an unknown header extension but that the SUT still accepts the RTP payload of those RTP packets.

Status: Mandatory

Preconditions: 1) The SUT is set to not use RTP header extensions during the test session, e.g. pre-configured or through no "a=extmap" lines being included in SDP signalling from the test instrument.
2) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling.
3) Data injection is set to use RTP header extension (RTP header X field set to 1) in some or all RTP packets during the test session, even though not announced through test instrument signalling. Any RTP header extension content according to section 5.3.1 of RFC 3550 that is suitable to the test instrument may be chosen, but the RTP header extension length field must be set larger than zero and the amount of RTP header extension data included in the RTP packet before the RTP payload must correspond to that length in each RTP packet.

Test procedure: 1) Observe the SUT application-level output based on RTP packet payload, e.g. audio from a speaker and/or video on a screen.
2) RTP packets with RTP header extension and RTP payload are sent from data injection.

Stop condition: Ten or more seconds have passed after first RTP packet was sent from data injection.

Pass criteria: Application-level output from the SUT exist as expected (e.g. audio and/or video) and is undistorted compared to a corresponding test setup where RTP header extension is not used.

Comments: The test result is agnostic to RTP packets being sent by the SUT or not.

6.3.4 RTP Contributing Source Tests

6.3.4.1 Ignore Contributing Source Test

Purpose:	Test if the SUT correctly ignores the CSRC list when receiving RTP packets with a non-zero CC field and CSRC list but that the SUT still accepts the RTP payload of those RTP packets.
Status:	Mandatory
Preconditions:	1) The SUT is set to receive RTP packets during the test session, e.g. pre-configured or set as send-receive or receive-only through test instrument signalling. 2) Data injection is set to use a CSRC list in some or all RTP packets during the test session (RTP header CC > 0). Any CSRC content suitable to the test instrument may be chosen, but the number of 32-bit CSRC values included in the CSRC list must correspond to the CC field in each RTP packet.
Test procedure:	1) Observe the SUT application-level output based on RTP packet payload, e.g. audio from a speaker and/or video on a screen. 2) RTP packets with CSRC list and RTP payload are sent from data injection.
Stop condition:	Ten or more seconds have passed after first RTP packet was sent from data injection.
Pass criteria:	Application-level output from the SUT exist as expected (e.g. audio and/or video) and is undistorted compared to a corresponding test setup where CSRC list is not used.
Comments:	The test result is agnostic to RTP packets being sent by the SUT or not.

6.4 SRTCP Tests

6.4.1 General

This clause describes tests that are only applicable to SRTCP but not to RTCP. Tests that are applicable to both SRTCP and RTCP are described in clause 6.2.

6.5 SRTP Tests

6.5.1 General

This clause describes tests that are only applicable to SRTP but not to RTP. Tests that are applicable to both SRTP and RTP are described in clause 6.3.

7 Conformance Indication

7.1 General

An RTP/RTCP implementation conforming to this specification should announce its compliance during call setup. Announcing compliance to this specification must use an SDP attribute "a=3gpp-rtp" (see clause 7.2).

7.2 The a=3gpp-rtp SDP attribute

7.2.1 General

The a=3gpp-rtp attribute may be provided on SDP session level, SDP media level or both. It is not defined for use on source or dcsa level. If used on SDP session level, it must be taken as applicable to all m= lines in the SDP using any RTP profile, e.g. RTP/AVP, RTP/AVPF, RTP/SAVP, or RTP/SAVPF. If used on SDP media level using any RTP profile, it must be taken as applicable only to that media description. It is not defined for use on non-RTP SDP media level.

7.2.2 ABNF syntax and semantics

The 3gpp-rtp ABNF syntax is:

3gpp-rtp-value = [3gpp-rtp-extension]

3gpp-rtp-extension = byte-string

; SP and byte-string as defined by IETF RFC 4566

Example:

a=3gpp-rtp

The optional attribute value is defined to allow for future extensibility and has currently no defined use.

7.2.3 SDP offer/answer considerations

This attribute is a unilateral announcement from the party creating the SDP offer or answer, and there are thus no SDP offer/answer considerations. When announcing compliance to this specification, the a=3gpp-rtp attribute shall be included in the SDP offer, and shall be included in the SDP answer regardless if it was also present in the corresponding SDP offer.

The optional attribute value should not be included in an SDP offer or answer sent by implementers of this specification and shall be ignored if received in an SDP offer or answer.

7.2.4 IANA registration information

Contact name, email address, and telephone number:

3GPP Specifications Manager

3gppContact@etsi.org

+33 (0)492944200

Attribute Name (as it will appear in SDP):

3gpp-rtp

Long-form Attribute Name in English:

3GPP-conformant RTP/RTCP implementation

Type of Attribute:

Session level and media level

Is Attribute Value subject to the Charset Attribute?

This attribute is not dependent on charset.

Purpose of the attribute:

This attribute is used to indicate the RTP/RTCP stack conformance to 3GPP TS 26.139.

Appropriate Attribute Values for this Attribute:

See 3GPP TS 26.139 clause 7.2.2 for ABNF and detailed usage.

MUX Category for this Attribute:

IDENTICAL

Annex A (informative):
Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2019-12	SA#86	SP-190991				Presented to TSG SA#86 (for information)	1.0.0
2020-03	SA#87-e	SP-200047				Presented to TSG SA#87-e (for approval)	2.0.0
2020-03	SA#87-e	SP-200047				Aproved by TSG SA#87-e	16.0.0

History

Document history		
V16.0.0	November 2020	Publication