**3GPP TSG-S4 Meeting #128*****S4-240974***

**Jeju, Republic of Korea, 20th–24th May 2024**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.0* | | | | | | | | |
| **PSEDUO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.113** | **CR** | **—** | **rev** | **—** | **Current version:** | **1.2.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | [iRTCW] QoE metrics reporting schema corrections | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | BBC | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | iRTCW | | | | |  | ***Date:*** | | | 2024-05-14 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **—** |  | | | | | ***Release:*** | | | Rel-18 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Errors and inconsistencies in the XML reception reporting schema for RTC. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Modify the schema to correct the errors and inconsistencies.   * uppercase initial for element names. * lowercase initial for attribute names. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The XML schema is not valid. | | | | | | | | |
| ***Q*** | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 15.2.2, 15.2.3, 15.2.4, 15.2.5, 15.2.6, 15.2.7, 15.2.8, 15.3.2 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | |  | | |
| ***affected:*** | |  | **X** | Test specifications | | | |  | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | |  | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | | pCR [S2-240974]: Submitted for WG agreement. | | | | | | | | |

## 

First change

# 2 References

[29] 3GPP TS 26.247: "Transparent end-to-end Packet-switched Streaming Services (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

Next change

### 15.2.2 Corruption duration metric

Corruption duration, M, is the time period from the NPT time of the last good frame (since the NPT time for the first corrupted frame cannot always be determined) before the corruption, to the NPT time of the first subsequent good frame. A corrupted frame may either be an entirely lost frame, or a media frame that has quality degradation and the decoded frame is not the same as in error-free decoding.

A good frame is a completely received frame:

- where all parts of the image are guaranteed to contain the correct content; or

- that is a refresh frame, that is, does not reference any previously decoded frames; or

- which only references previously decoded good frames

Completely received means that all the bits are received and no bit error has occurred.

Corruption duration, M, in milliseconds can be calculated as below:

a) M can be derived by the client using the codec layer, in which case the codec layer signals the decoding of a good frame to the client. A good frame could also be derived by error tracking methods, but decoding quality evaluation methods shall not be used.

b) Alternatively, the corruption is considered as ended after N milliseconds with consecutively completely received frames, or when a refresh frame has been completely received, whichever comes first.

The optional configuration parameter N can be set to define the average characteristics of the codec. If N has not been configured it shall default to the length of one measurement interval for video media, and to one frame duration for non-video media.

The N parameter is specified in milliseconds and is used with the "CorruptionDuration" parameter. The value of N may be set by the server.

All the occurred corruption durations within each measurement period are summed and stored in the vector @*totalCorruptionDuration*. The unit of this metrics is expressed in milliseconds. Within each measurement period the number of individual corruption events are summed up and stored in the vector @*numberOfCorruptionEvents.*

The syntax for the metric "CurruptionDuration" is as defined in Table 15.2.2-1

Table 15.2.2-1: Corruption duration metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| CorruptionDuration | | Object |  |
|  | @totalCorruption‌Duration | unsignedLongVectorType | An unordered list of all occurred corrupt durations within each measurement period. |
|  | @numberOf‌CorruptionEvents | unsignedLongVectorType | An unordered list of corruption events occurred within each measurement period. Within each measurement period the number of individual corruption events are summed up and stored. |

### 15.2.3 Successive loss of RTP packets

The metric "SuccessiveLoss" indicates the number of RTP packets lost in succession per media channel.

All the number of successively lost RTP packets are summed up within each measurement resolution period of the stream and stored in the vector @*totalNumberOfSuccessivePacketLoss*. The unit of this metric is expressed as an integer equal to or larger than 0. The number of individual successive packet loss events within each measurement resolution period are summed up and stored in the vector @*numberOfSuccessiveLossEvents.* The number of received packets are also summed up within each measurement resolution period and stored in the vector @*numberOfReceivedPackets.* These three vectors are reported by the RTC UE as part of the QoE report.

The syntax for the metric "SuccessiveLoss" is as defined in Table 15.2.3-1.

Table 15.2.3-1: Successive loss of RTP packets metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| SuccessiveLoss | | Object |  |
|  | @totalNumberOf‌Successive‌Packet‌Losses | unsignedLongVectorType | An unordered list of all successively lost RTP packets within each measurement period. |
|  | @numberOf‌Successive‌Loss‌Events | unsignedLongVectorType | The number of individual successive packet loss events within each measurement resolution period are summed up and stored in the vector. Provides an unordered list of successive packet loss events (occurred within each measurement period) measured during a metric reporting period. |
|  | @numberOf‌Received‌Packets | unsignedLongVectorType | The number of received packets are summed up within each measurement resolution period and stored in the vector. |

### 15.2.4 Media frame rate

Frame rate indicates the media playback frame rate. The playback frame rate is equal to the number of frames displayed during the measurement resolution period divided by the time duration, in seconds, of the measurement resolution period.

For the Metrics-Name "MediaFrameRate", the value field indicates the frame rate value. This metric is expressed in frames per second and can be a fractional value. The frame rates for each resolution period are stored in the vector *MediaFrameRate* and reported by the RTC UE as part of the QoE report.

The syntax for the metric "MediaFrameRate" metric is as defined in Table 15.2.4-1.

Table 15.2.4-1: Media frame Rate metric for Quality Reporting

|  |  |  |
| --- | --- | --- |
| Key | Type | Description |
| MediaFrameRate | doubleVectorType | An unordered list of media frame rate values reported over a reporting period. The frame rates for each metric resolution period are stored in the vector. |

### 15.2.5 Jitter duration

Jitter happens when the absolute difference between the actual playback time and the expected playback time is larger than *Jitterthreshold* in milliseconds. The expected time of a frame is equal to the actual playback time of the last played frame plus the difference between the NPT time of the frame and the NPT time of the last played frame.

The optional configuration parameter *Jitterthreshold* can be set to control the amount of allowed jitter. If the parameter has not been set, it defaults to 100 ms. The *Jitterthreshold* parameter is specified in milliseconds and is used with the "JitterDuration" parameter. The value of *Jitterthreshold* may be set by the server.

All the jitter durations are summed up within each measurement resolution period and stored in the vector @*totalJitterDuration*. The unit of this metric is expressed in seconds and can be a fractional value. The number of individual events within the measurement resolution period are summed up and stored in the vector @*numberOfJitterEvents.* These two vectors are reported by the RTC UE as part of the QoE report.

The syntax for the metric "JitterDuration" is as defined in Table 15.2.5-1.

Table 15.2.5-1: Jitter duration metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| JitterDuration | | Object |  |
|  | @totalJitterDuration | doubleVectorType | All the jitter durations are summed up within each measurement resolution period and stored in the vector. |
|  | @numberOfJitterEvents | unsignedLongVectorType | The number of individual events within the measurement resolution period are summed up and stored in the vector. Provides An unordered list of jitter events (occurred within each measurement period) measured during a metric reporting period. |

### 15.2.6 Sync loss

Sync loss happens when the absolute difference between value A and value B is larger than *SyncThreshold* in milliseconds. Value A represents the difference between the playback time of the last played frame of the video stream and the playback time of the last played frame of the speech/audio stream. Value B represents the difference between the expected playback time of the last played frame of the video stream and the expected playback time of the last played frame of the speech/audio stream.

The optional configuration parameter s*yncthreshold* can be set to control the amount of allowed sync mismatch. If the parameter has not been set, it defaults to 100 ms. The s*yncthreshold* parameter is specified in milliseconds and is used with the "SyncLoss" parameter. The value of *syncthreshold* may be set by the server.

All the sync loss durations are summed up within each measurement resolution period and stored in the vector @*totalSyncLossDuration*. The unit of this metric is expressed in seconds and can be a fractional value. The number of individual events within the measurement resolution period are summed up and stored in the vector @*numberOfSyncLossEvents.* These two vectors are reported by the RTC UE/endpoint as part of the QoE report.

The syntax for the metric "SyncLoss" is as defined in Table 15.2.6-1.

Table 15.2.6-1: Sync loss metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| SyncLoss | | Object |  |
|  | @totalSyncLossDuration | doubleVectorType | All the sync loss durations are summed up within each measurement resolution period and stored in the vector. |
|  | @numberOfSyncLossEvents | unsignedLongVectorType | The number of individual sync loss events within the measurement resolution period are summed up and stored in the vector. Provides an unordered list of sync loss events (occurred within each measurement period) measured during a metric reporting period. |

### 15.2.7 [Round-trip](https://www.rfc-editor.org/rfc/rfc8834#name-temporal-spatial-trade-off-) time

The round-trip time (RTT) consists of the RTP-level round-trip time, plus the additional two-way delay due to buffering and other processing in each RTC UE.

The last RTCP round-trip time value estimated during each measurement resolution period shall be stored in the vector @*networkRTT*. The unit of this metrics is expressed in milliseconds.

The two-way additional internal client delay valid at the end of each measurement resolution period shall be stored in the vector @*internalRTT*. The unit of this metrics is expressed in milliseconds.

The two vectors are reported by the RTC UE as part of the QoE report.

The syntax for the metric "RoundTripTime" is as defined in Table 15.2.7-1.

Table 15.2.7-1: Round-trip time metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| RoundTripTime | | Object |  |
|  | @networkRTT | unsignedLongVectorType | The last RTCP round-trip time value estimated during each measurement resolution period shall be stored in the vector. |
|  | @internalRTT | unsignedLongVectorType | The two-way additional internal client delay valid at the end of each measurement resolution period shall be stored in the vector. |

### 15.2.8 Average bit rate

The average codec bit rate is the bit rate used for coding "active" media information during the measurement resolution period.

For speech media the average codec bit rate can be calculated as the number of "active" speech bits received for "active" frames divided by the total time, in seconds, covered by these frames. The total time covered is calculated as the number of "active" frames times the length of each speech frame.

For non-speech media the average codec bit rate is the total number of RTP payload bits received, divided by the length of the measurement resolution period.

The average codec bit rate value for each measurement resolution period shall be stored in the vector @*averageCodecBitRate*. The unit of this metrics is expressed in kbit/s and can be a fractional value. The vector is reported by the RTC UE/endpoint as part of the QoE report.

The syntax for the metric "AverageBitRate " is as defined in Table 15.2.8-1.

Table 15.2.8-1: Average bit rate metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| AverageBitRate | | Object |  |
|  | @averageCodec‌BitRate | doubleVectorType | The average codec bit rate value for each measurement resolution period shall be stored in the vector. |

Next change

### 15.3.2 Report format

The QoE report is formatted as an XML document that complies with the XML schema in listing 10.6.2‑1 of TS 26.247 [29].

The schema in listing 15.3.2‑1 is an extension to allow additional QoE metrics for RTC to be reported using the QoE report specified in clause 10.6.2 of TS 26.247 [29].

Listing 15.3.2-1: QoE Metrics XML schema

|  |
| --- |
| <?xml version="1.0"?> <xs:schema version="TSG104-Rel18" xmlns:xs="http://www.w3.org/2001/XMLSchema"  targetNamespace="urn:3gpp:metadata:2023:RTC:QoEMetrics"  xmlns:sv="urn:3gpp:metadata:2016:PSS:schemaVersion"  xmlns="urn:3gpp:metadata:2023:RTC:QoEMetrics" elementFormDefault="qualified">  <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>  <xs:element name="QoeMetric" type="QoeMetricType"/>    <xs:complexType name="QoeMetricType">  <xs:sequence>  <xs:choice>  <xs:element name="CorruptionDuration" type="CorruptionDurationType"/>  <xs:element name="SuccessiveLoss" type="SuccessiveLossType"/>  <xs:element name="MediaFrameRate" type="doubleVectorType"/>  <xs:element name="JitterDuration" type="JitterDurationType"/>  <xs:element name="SyncLoss" type="SyncLossType"/>  <xs:element name="RoundTripTime" type="RoundTripTimeType"/>  <xs:element name="AverageBitRate" type="AverageBitRateType"/>  </xs:choice>  <xs:element ref="sv:delimiter"/>  <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="CorruptionDurationType">  <xs:attribute name="totalCorruptionDuration" type="unsignedLongVectorType" use="required"/>  <xs:attribute name="numberOfCorruptionEvents" type="unsignedLongVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="SuccessiveLossType">  <xs:attribute name="totalNumberOfSuccessivePacketLosses" type="unsignedLongVectorType" use="required"/>  <xs:attribute name="numberOfSuccessiveLossEvents" type="unsignedLongVectorType" use="required"/>  <xs:attribute name="numberOfReceivedPackets" type="unsignedLongVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="JitterDurationType">  <xs:attribute name="totalJitterDuration" type="doubleVectorType" use="required"/>  <xs:attribute name="numberOfJitterEvents" type="unsignedLongVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="SyncLossType">  <xs:attribute name="totalSyncLossDuration" type="doubleVectorType" use="required"/>  <xs:attribute name="numberOfSyncLossEvents" type="unsignedLongVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="RoundTripTimeType">  <xs:attribute name="networkRTT" type="unsignedLongVectorType" use="required"/>  <xs:attribute name="internalRTT" type="unsignedLongVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="AverageBitRateType">  <xs:attribute name="averageCodecBitRate" type="doubleVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:simpleType name="unsignedLongVectorType">  <xs:list itemType="xs:unsignedLong"/>  </xs:simpleType>  <xs:simpleType name="doubleVectorType">  <xs:list itemType="xs:double"/>  ....</xs:simpleType>  <xs:simpleType name="StringVectorType">  <xs:list itemType="xs:string"/>  </xs:simpleType>  <xs:simpleType name="UnsignedIntVectorType">  <xs:list itemType="xs:unsignedInt"/>  </xs:simpleType>  </xs:schema> |

End of changes