**3GPP TSG-WG SA4 Meeting #121 *S4-221366***

**Toulouse, France, November 14 – 18, 2022**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  | **26.114** | **CR** | **0532** | **rev** | **-** | **Current version:** | **17.5.0** |  |
|  |
| *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 | **X** | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | CR to TS 26.114 Add slice scope into the QoE configuration  |
|  |  |
| ***Source to WG:*** | Huawei, HiSilicon |
| ***Source to TSG:*** | SA4 |
|  |  |
| ***Work item code:*** | NR\_QoE\_enh-Core |  | ***Date:*** | 2022-11-04 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-17 |
|  | *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 canbe 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-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** |

|  |
| --- |
| In the LS R3-225227 from RAN3, it’s asked to introduce the slice scope information in the QoE configuration container to support the per-slice QoE reporting. In current SA4 specs, the slice scope is missing in the QoE configuration container only with mandatory dnn/slice information in the report, which may lead to unnecessary QoE collection and reporting.  |

 |
|  |  |
| ***Summary of change:*** | Add slice scope in the QoE metric configuration to indicate whether the QoE metric collection and reporting should be executed on the specific network slices. |
|  |  |
| ***Consequences if not approved:*** | Unalignment between SA4 and RAN3. |
|  |  |
| ***Clauses affected:*** | 16.4, 16.5.2 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

\* \* \* \* First change \* \* \* \*

## 16.4 Metrics Reporting

When a session is started, the MTSI client must determine whether QoE reporting is required for the session. If the parameter "Enabled" is set to false, no QoE reporting shall be done. If the "Enabled" parameter is set to true the optional "Rules" parameters are checked (sub-clause 16.3.3) to define if QoE reporting shall be done.

Once the need for QoE reporting has been established, the client shall continuously compute all specified metrics for each measurement interval period, according to the "Measure-Resolution" parameter (sub-clause 16.3.2). In order to bound the resources used by metrics reporting, the minimum values for the Measure-Resolution and Sending-Rate are specified to be 5 seconds and 30 seconds respectively. The computed metrics are represented in a vector format, adding an additional metric value to each metric vector after each new measurement interval period.

Note that the calculated metrics shall only cover one measurement interval. For instance, if the corruption duration extends longer than to the end of the current measurement interval, only the portion which fits into the current measurement interval shall be reported. The remaining portion of the corruption duration shall be reported as belonging to the next measurement interval.

The end of the session will normally not correspond to the end of a measurement interval period, so the metrics for the last measurement interval period will typically be calculated over a time shorter than the configured measurement interval. Note, however, that these last metrics shall still be added to the metrics vectors and reported to the server.

It is possible for the server to use the start and stop timestamps, together with the knowledge of the configured measurement interval, to derive the actual length of the last measurement interval period, but any specific action or interpretation of these last shorter measurements is out of scope of this specification.

The MTSI client shall send QoE report messages to the server in accordance with the specified reporting interval "Sending-Rate" (sub-clause 16.3.2). All stored metrics data shall then be sent to the server, and then deleted from the metrics storage.

Note that if the reporting interval is not an integer multiple of the measurement interval, only the measurement interval periods which have been fully passed shall be included in the report. The ongoing not-passed measurement interval period shall be included in the next report. The only exception is at the end of the session, where also the last ongoing measurement interval period shall be directly calculated and included in the report.

If QoE configuration has been done via the OMA MO, the client shall send QoE reports using the HTTP (RFC 2616 [73]) POST request carrying XML formatted metadata. If the optional "APN" parameter is defined in the OMA managed object, that APN shall be used for establishing the PDP context or EPS bearer on which the QoE metric reports will be transmitted. The MTSI client randomly selects one of the URIs from the MO "Server" parameter, with uniform distribution.

If QoE configuration has been done via the QMC functionality (see clause 16.5), the client shall also send the QoE reports as described in clause 16.5. Note that for QMC scheme, if the SliceScope is included in the QoE configuration and the slice associated with the MTSI service is within the SliceScope, the QoE collection shall be executed and the S-NSSAI and DNN that correspond to the report data shall be included for support of per-slice QoE reporting and evaluation in OAM. This information may be retrieved via the AT Command +CGDCONT [161]) or the specific traffic mapping with URSP rule [182].

Each QoE report is formatted in XML according the following XML schema (sub-clause 16.4.1). An informative example of a single reception report XML object is also given (sub-clause 16.4.2). The reports should be compressed using GZIP only if the MO parameter "Format" specifies this.

Each QoE Metrics element has a set of attributes and any number of media level QoE Metrics elements. All attributes are defined in sub-clause 16.4.1 and correspond to the QoE metrics listed in sub-clause 16.2. Individual metrics can be selected as described in sub-clause 16.3.2.

Except for the media level QoE metrics, the following parameters shall be reported for each report:

- The *callId* attribute identifies the call identity of the SIP session.

- The *clientId* attribute is unique identifier for the receiver, e.g. an MSISDN of the UE as defined in [80].

- The *startTime* and *stopTime* attributes identifies the client NTP time when the measurements included in the report were started and stopped. The time is based on the local real-time clock in the client, and might not be consistent with the true NTP time. However, assuming that the reporting is done without any extra delay the server can use the *stopTime* attribute to correct the timestamps if necessary.

- The *mediaId* attribute shall be reported for each media level QoE report, and identifies the port number for the media.

 If the attribute *qoeReferenceId* was defined in the QMC configuration (see clause 16.5.2), the value shall be copied into each QoE report, to facilitate network-side correlation (see [178]). If this attribute was defined, the attribute *recordingSessionId* shall also be returned for each QoE report. The *recordingSessionId* is a two-byte octet defined by the client. It shall remain the same for all QoE reports belonging to the same session, and it should be different for QoE reports belonging to different sessions.

\* \* \* \* Second change \* \* \* \*

### 16.5.2 XML configuration

When QoE reporting is configured via the QMC functionality, the configuration basically contains the same information as in the QoE metrics reporting managed object (see clause 16.3.1), but encapsulated according to the XML scheme below. Note that the managed object leaves "Servers", "APN" and "Format" are not needed for the QMC functionality, and thus not included.

Note that if geographical filtering is handled on the network side (i.e. QoE reporting is turned on/off by the network depending on the UE location), no LocationFilter should be specified in the QoE Configuration, as this would mean two consecutive filterings.

Also note that the optional attribute qoeReferenceId is a reference set by the network side (see [178]), which is not directly used by the client. However, if this attribute is defined, it shall be copied into each QoE report, to facilitate network-side correlation.

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:3gpp:metadata:2017:MTSI:qoeconfig"

 elementFormDefault="qualified"

 xmlns:xs="http://www.w3.org/2001/XMLSchema"

 xmlns:sv="urn:3gpp:metadata:2017:MTSI:schemaVersion"

 xmlns="urn:3gpp:metadata:2017:MTSI:qoeconfig">

 <xs:element name="MTSIQualityReporting" type="QualityReportingType"/>

 <xs:complexType name="QualityReportingType">

 <xs:sequence>

 <xs:element name="LocationFilter" type="LocationFilterType" minOccurs="0"/>

 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:attribute name="enabled" type="xs:boolean" use="required"/>

 <xs:attribute name="rules" type="xs:string" use="optional"/>

 <xs:attribute name="speechMetrics" type="xs:string" use="optional"/>

 <xs:attribute name="videoMetrics" type="xs:string" use="optional"/>

 <xs:attribute name="textMetrics" type="xs:string" use="optional"/>

 <xs:attribute name="SliceScope" type="UnsignedIntVectorType" use="optional"/>

 <xs:attribute name="qoeReferenceId" type="xs:hexBinary" use="optional"/>

 <xs:anyAttribute namespace="##other" processContents="lax"/>

 </xs:complexType>

 <xs:complexType name="LocationFilterType">

 <xs:sequence>

 <xs:element name="cellID" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>

 <xs:element name="shape" type="ShapeType" minOccurs="0"/>

 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:anyAttribute namespace="##other" processContents="lax"/>

 </xs:complexType>

 <xs:complexType name="ShapeType">

 <xs:sequence>

 <xs:element name="PolygonList" type="PolygonListType" minOccurs="0"/>

 <xs:element name="CircularAreaList" type="CircularAreaListType" minOccurs="0"/>

 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:anyAttribute namespace="##other" processContents="lax"/>

 </xs:complexType>

 <xs:complexType name="PolygonListType">

 <xs:annotation>

 <xs:documentation> see [OMA MLP] </xs:documentation>

 </xs:annotation>

 <xs:sequence>

 <xs:element name="Polygon" minOccurs="0" maxOccurs="unbounded"/>

 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:attribute name="ConfLevel" type="xs:unsignedInt" use="optional"/>

 <xs:anyAttribute namespace="##other" processContents="lax"/>

 </xs:complexType>

 <xs:complexType name="CircularAreaListType">

 <xs:annotation>

 <xs:documentation> see [OMA MLP] </xs:documentation>

 </xs:annotation>

 <xs:sequence>

 <xs:element name="CircularArea" minOccurs="0" maxOccurs="unbounded"/>

 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:attribute name="ConfLevel" type="xs:unsignedInt" use="optional"/>

 <xs:anyAttribute namespace="##other" processContents="lax"/>

 </xs:complexType>

 <xs:simpleType name="UnsignedIntVectorType">
 <xs:list itemType="xs:unsignedInt"/>
 </xs:simpleType>

</xs:schema>

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:3gpp:metadata:2017:MTSI:schemaVersion"

 xmlns="urn:3gpp:metadata:2017:MTSI:schemaVersion"

 xmlns:xs="http://www.w3.org/2001/XMLSchema"

 elementFormDefault="qualified">

 <xs:element name="schemaVersion" type="xs:unsignedInt"/>

 <xs:element name="delimiter" type="xs:byte"/>

</xs:schema>

\* \* \* \* End of changes \* \* \* \*