**3GPP TSG SA WG4 Meeting #128*****S4-240xxx***

**Jeju, Korea, 20th–24th May 2024**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **26.506** | **CR** | **00xx** | **rev** | **0** | **Current version:** | **18.2.0** |  |
|  |
| *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:***  | Terminology alignment in RTC architecture |
|  |  |
| ***Source to WG:*** | Samsung, BBC |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** |  GA4RTAR |  | ***Date:*** | 2024-04-25 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***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 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-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | In developing generic architecutre and APIs between 5GMS and RTC, it was raised that some of names and terminologies should be rephased. In particular, the term RTC endpoint is indiscreetly used to indicate the function both in UE and RTC AS. The protocol requirements should be different between RTC endpoint in UE and another in RTC AS, so this should be distinguished. In addition, the existing RTC architecture does not sufficiently address the peer-to-peer media transport case, which is essntial.  |
|  |  |
| ***Summary of change:*** | * Provides more detailed definition for architecture functions: RTC endpoint, RTC Client, RTC Access Function, WebRTC Framework
* Adds missing reference points: RTC-10 (AS to AS), RTC-12 (peer-to-peer)
* Updates relevant text according to re-specified definition
 |
|  |  |
| ***Consequences if not approved:*** | Inconsistent terminologies between 5GMS (as specified in TS 26.501 and 26.510) and RTC architecture |
| ***Q*** |  |
| ***Clauses affected:*** | 3.1, 4.1.1, 4.1.2.3, 4.1.2.4, 4.2.4, 4.2.6, 4.2.7, 4.2.8, 4.2.9, 4.2.10, 4.2.11, 4.3.3, 4.3.6, 4.3.8, 4.3.9 (new), 4.3.10 (new), 4.4.2.1, 5.4, 5.5, 6.1, 6.2 |
|  |  |
|  | **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:*** |  |

FIRST CHANGE

## 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].

**RTC endpoint:** An instance of the WebRTC Framework capable of participating in an RTC session deployed either in the RTC Access Function of a UE or in the Media Function of an RTC AS.

**RTC Client:** UE function comprising an RTC Access Function and a RTC Media Session Handler which interacts with functions in the network and UE applications.

**RTC Access Function:** A set of functions including an instance of the WebRTC framework that exchanges real-time media with one or more RTC endpoints via reference point RTC-4 or RTC-12, and that exposes client APIs defined in the present document to the Native WebRTC App at reference point RTC-7 and to the RTC Media Session Handler at reference point RTC-11, as well as exposing the W3C-defined WebRTC API to the web application running in a browser.

**WebRTC Framework:** A well-defined subset of the WebRTC protocol stack that supports real-time communication of media between an RTC endpoint and its peer(s).

Next change

## 4.1 Overall architecture for Real-Time media Communication (RTC)

### 4.1.1 Definition of RTC architecture

Real-Time media Communication (RTC) over 5G system in the context of this specification is defined as the delivery of delay-sensitive media from one peer to another with support of 5G network. AR conversational service described in TR 26.998 [2] is a typical use cases for RTC, which enables end-users to directly communicate real-time media including AR/MR media content as specified in TS 26.119 [3]. As identified in clause 8.4 of TR 26.998, there may be different options to enable such AR conversational service, for example re-use of parts of MTSI as defined in TS 26.114 [10] such as the IMS data channel or 5G Media Streaming for managed services.

The overall RTC architecture is shown in Figure 4.1.1-1 as below.



Figure 4.1.1-1: Real-time media communication (RTC) in 5G System

NOTE: The functions indicated by the yellow filled boxes are in scope of the present document for RTC. The functions indicated by the grey boxes are defined in 5G System specifications. The functions indicated by the blue boxes are neither in scope of 5G RTC nor 5G System specifications.

The media data is exchanged between two or more RTC Clients over a 5G System as defined in TS 23.501 [11]. The RTC endpoint is an instance of the WebRTC Framework configured by the RTC System defined in the present document. An RTC endpoint is typically realised by the RTC Client of a UE, but an RTC AS, including an edge computing server may also play the role of RTC endpoint, as defined in clause 4.4.2. The Application Provider provides a RTC Aware-Application on the UE to make use of RTC endpoint and network functions using interfaces and APIs. RTC architecture provides the core functions and entities to support WebRTC-based service over 5G System, two main functions are defined in the trusted DN.

- RTC AF: An Application Function as defined in TS 26.501 [6], but dedicated to real-time media communication.

- RTC AS: An Application Server dedicated to real-time media communication.

NOTE: If both the RTC AF and RTC AS are deployed in an external DN, this is out of scope of the present document.

The detailed RTC architecture mapping to the overall high-level architecture in Figure 4.1.1-1 is shown in Figure 4.1.1-2 below.

 

Figure 4.1.1-2: RTC General Architecture

NOTE 1: Some subfunctions may not be required depending on the collaboration scenario. Description of collaboration scenario and its architecture variant are specified in annex A.

NOTE 2: The WebRTC Framework is a WebRTC protocol stack whose implementation is specified by W3C and IETF.

NOTE 3: Red ovals indicate API provider functions.

NOTE 4: The RTC Access Function may be realised by a web browser in deployments of the RTC Client that support web-based applications through the WebRTC API.

The WebRTC Signalling Function may be co-located with the RTC AF. In such deployments, the WebRTC Signalling Function acts as an RTC AF with access to the 5G Core, and some of the RTC AF interactions with the WebRTC Signalling Function may be replaced to avoid concurrent/redundant requests from the RTC endpoint in the UE. Specifically, media session handling interactions between the RTC AF and the UE at reference point RTC‑5 may be replaced by the equivalent WebRTC signalling interactions defined at reference point RTC‑4.

The subfunctions inside the RTC AF, RTC AS and the RTC Client are defined in clause 4.2 and the reference points shown in Figure 4.1.1-2 are defined in clause 4.3.

### 4.1.2 Generalized Media Delivery architecture

#### 4.1.2.1 Generalized Media Delivery in the 5G System

This clause and subsequent subclauses of clause 4.1.2 define a generalized Media Delivery architecture of which the architecture for Real-Time Communication (RTC) defined elsewhere in the present document is one possible realisation. In case of any misalignment between the two, the RTC architecture has precedence over this generalised architecture.

Due to the similarity of the 5GMS architecture (as defined in TS 26.501 [6]) to the architecture for Real-Time media Communication (RTC) defined in the present document, the RTC functions and 5GMS functions may share or may make use of many common functionalities for both media session handling and media delivery. A generalized Media Delivery architecture that integrates 5GMS and RTC functionality in the 5G System is defined in figure 4.1.2.1-1.

NOTE: Full integration of 5GMS and RTC is not addressed in the present document.



Figure 4.1.2.1-1: Generalized Media Delivery architecture within the 5G System

In this representation:

- The *Media Application Provider* plays the role of the RTC Application Provider.

- The *Media-aware Application* plays the role of the Native WebRTC App.

- The RTC AF is one possible realisation of the general *Media AF*.

- The RTC AS is one possible realisation of the general *Media AS*.

- The RTC endpoint is part of the general *Media Client*.

#### 4.1.2.2 Reference architecture for Media Delivery

A functional description with additional details as well as reference points is provided below, as illustrated in figure 4.1.2.2-1.



NOTE 1: Exposed APIs are named in *italics*.

NOTE 2: If the Media Client is deployed as a monolithic functional block, it may choose not to expose interfaces externally at reference point M11.

Figure 4.1.2.2-1: Generalized Media Delivery architecture

#### 4.1.2.3 Network Functions and UE entities

Functional definitions may be generalized as follows:

- **Media AF:** An Application Function as defined in clause 6.2.10 of TS 23.501 [11] dedicated to Media Delivery.

- **Media AS:** An Application Server dedicated to Media Delivery.

- **Media Client:** A UE internal function dedicated to Media Delivery comprising:

- **Media Session Handler:** An entity on the UE that communicates with the Media AF in order to establish, control and support the delivery of a media session.

- **Media Access Function:** An entity on the UE that communicates with the Media AS in order to access and deliver media content. The media access function for example may be further sub-divided into content delivery protocols, codecs, media types and metadata representation.

- **Media-aware Application:** An application entity on the UE that makes use of 3GPP-defined APIs to invoke the Media Session Handler and/or the Media Access Function in order to support Media Delivery.

NOTE: An application (e.g., a web browser application) that does not invoke either the Media Session Handler or the Media Access Function using 3GPP-defined APIs is not considered a Media-aware Application and is not mapped into the generalized Media Delivery reference architecture.

Table 4.1.2.3-1: Mapping of RTC functions to generalized Media Delivery architecture

|  |  |
| --- | --- |
| Generalized media architecture function | RTC function |
| Media AF | RTC AF |
| Media AS | RTC AS |
| Media Client | RTC Client |
|  | Media Session Handler | RTC Media Session Handler |
|  | Media Access Function | RTC Access Function |
| Media Application Provider | RTC Application Provider |
| Media-aware Application | Native WebRTC App |

#### 4.1.2.4 Reference points

The following reference points are defined for Media Delivery:

**M1**: Reference point between the Media Application Provider and the Media AF for the provisioning of Media Delivery.

**M2**: Reference point between the Media Application Provider and the Media AS for the purposes of ingesting media into the Media AS or egesting media from the Media AS.

NOTE 1: Reference point M2 is not defined by the RTC architecture in this release.

**M3**: Reference point between the Media AF and the Media AS for the purposes of Media AS configuration and/or for media session handling in relation to Media Delivery.

NOTE 2: Reference point M3 is defined by the RTC architecture in this release but specification is for future study.

**M4**: Reference point between the Media AS and the Media Access Function in the UE for the purpose of downlink transport of media from the Media AS to the Media Access Function ("content distribution") or uplink transport of media from the Media Access Function to the Media AS ("content contribution").

NOTE 3: Session setup signalling at reference point RTC‑4 lies outside the scope of reference point M4.

**M5**: Reference point between the Media AF and the Media Session Handler in the Media Client for the purpose of media session handling in relation to Media Delivery.

**M6**: Reference point between the Media-aware Application and the Media Session Handler for the purpose of configuring the Media Session Handler.

**M7**: Reference point between the Media-aware Application and the Media Access Function for the purpose of media access control.

**M8**: Reference point between the Media-aware Application and the Media Application Provider.

NOTE 4: Reference point M8 is private and therefore beyond the scope of standardisation.

**M9**: Reference point between one instance of the Media AF and another for the purpose of Media AF instance chaining.

NOTE 5: Reference point M9 is not defined by the RTC architecture.

**M10**: Reference point between one instance of the Media AS and another for the purpose of distributed service chaining over multiple Media ASs.

NOTE 6: Reference point M10 is defined but not further specified by the present document in this release.

**M11**: Reference point between the Media Session Handler and the Media Access Function (both in the Media Client) for the purpose of configuring the Media Session Handler and/or media access control.

**M12**: Reference point between one RTC Access Function in a UE and another for the purpose of peer-to-peer media transport between different Media Clients.

Table 4.1.2.4-1: Mapping of RTC reference points to generalized Media Delivery architecture

|  |  |
| --- | --- |
| Generalized Media Delivery architecture reference point | RTC reference point |
| M1 | RTC‑1 |
| M2 | Not defined |
| M3 | RTC‑3 |
| M4 | RTC‑4 |
| M5 | RTC‑5 |
| M6 | RTC‑6 |
| M7 | RTC‑7 |
| M8 | RTC‑8 |
| M9 | Not defined |
| M10 | RTC-10 |
| M11 | RTC‑11 |
| M12 | RTC-12 |

Next change

### 4.2.4 RTC Media Session Handler (MSH)

The RTC MSH is an entity running on the UE, which assists with the 5G integration of the WebRTC application. It exchanges, on behalf of the application, information about the WebRTC sessions with the network.

The RTC MSH receives information about a new WebRTC session from the application. It relays the information to the Network Support Function. It also receives events and other network information about the WebRTC session from the Network Support Function, which it may relay to the application.

In addition, one of subfunction in RTC MSH is the metric collection and reporting. It executes the collection of QoS and QoE metrics measurements from the RTC Access Function and the WebRTC application and sends metrics reports to the RTC AF for the purpose of metrics analysis or to enable potential transport optimizations by the network.

### 4.2.5 Network support function

The support functionality includes the following:

- Network Support Function receives information from the UE and/or other ASs about a WebRTC session and its state

- Network Support Function requests the network that QoS should be allocated (or satisfied) for a starting or modified session

- Network Support Function receives notification from the network about changes to the QoS allocation for the ongoing WebRTC session

- Network Support Function exchanges information about the WebRTC session with the trusted STUN/TURN/Signalling function, e.g. to identify a WebRTC session and associate it with a QoS template.

NOTE: The integration/collocation of this RTC AF and WebRTC signalling function is possible. Co-located WebRTC signalling function is able to act as a RTC AF which is accessible to 5GC, and replace some of this RTC AF’s interfaces and APIs with WebRTC signalling function. For example, interfaces and APIs between this RTC AF and UE will be replaced to avoid concurrent/redundant requests from UE.

### 4.2.6 ICE functions

The MNO may offer trusted ICE functions to the WebRTC application to be used during the WebRTC ICE gathering phase. These functions may be STUN and TURN servers that facilitate NAT and firewall traversal.

The MNO-operated trusted ICE functions may assist with the 5G integration of the WebRTC application. This could be done by triggering network assistance to starting or ongoing WebRTC sessions.

### 4.2.7 WebRTC signalling function

The trusted WebRTC signalling function is used to setup and manage MNO-operated WebRTC applications. They offer a standardized signalling protocol for the session setup to both parties of the WebRTC session. The WebRTC signalling function handles the offer/answer exchange and has access to the SDP in both directions.

The WebRTC signalling function may use that knowledge to offer network assistance and other 5G features to the endpoints of the WebRTC session.

The WebRTC signalling function manages media flow sessions in both uplink and downlink directions.

### 4.2.8 Interworking function

This function provides interworking functionality to enable MNO-facilitated WebRTC sessions that involve endpoints across different MNOs. They may for example provide cross-network signalling functionality to allow WebRTC signalling server that are hosted in different networks to communicate, in order to establish and manage the WebRTC sessions.

### 4.2.9 Transport gateway function

A transport gateway function may be offered by the MNO to support cross-operator WebRTC sessions. It may offer the border control function for user plane (e.g., topology hiding, IPv4-IPv6 translation) as a gateway, which is located at the network boundary where different operators or third-party network connects. It works under the control of the trusted interworking function.

Note: Detailed functionality is specified in TR 26.930 [5].

### 4.2.10 Media function

A media server may be offered by the MNO to support WebRTC sessions. It may offer a wide range of functionality such as:

- a content server that serves content to the WebRTC application, e.g. through a data channel

- media processing functionality: may be used by the WebRTC application as a relay that performs some media processing function such as transcoding, recording, 3D reconstruction, etc.

- scene composition functionality: the server may compose a 3D scene and distribute it to several point-to-point WebRTC sessions

- Multi-point Control Unit (MCU) functionality: the server may offer multi-party conferencing functionality to merge a number of point-to-point WebRTC sessions

- Selective Forwarding Unit (SFU) functionality: the server may offer the selection, copy, and forwarding functionality of IP steams produced by multiple RTC Clients (i.e., participants).

- Maintain uplink and downlink flow context (QoS, remote control and etc.) by interacting with the WebRTC signalling function.

### 4.2.11 Application supporting web function

A web server may be offered by the MNO to support applications by providing web service entry point, authorization/authentication, sharing files, or scheduling conferencing sessions.

Next change

### 4.3.3 RTC-4: Media-centric transport interface

This interface is used to exchange the WebRTC traffic between RTC Access Function and RTC AS as well as to exchange signalling information related to the WebRTC session with the trusted application servers.

The traffic includes:

- Media streams sent over RTP

- Application data sent over data channel

- WebRTC Signalling data along with STUN and TURN servers

- Other application data

RTC-4 may further be grouped into two sub-interfaces as follows.

**RTC-4s:**

The RTC-4s interface is an interface between the RTC Access Function and the RTC AS such as WebRTC Signalling Function. This interface is used for the exchange of signalling information related to the WebRTC session between two or more RTC endpoints using trusted application servers. In some cases where the signalling is not handled by RTC Access Function, the RTC-4s interface is an interface between the native WebRTC applications and the WebRTC Signalling server.

**RTC-4m:**

This interface is used for transmission of media and other related data between two or more RTC endpoints when, at least, one of RTC endpoints is instantiated in the RTC AS.

The traffic includes

- Media data transmitted over RTP

- Application data transmitted using Data channel

- Media related meta-data transmitted using Data channel

NOTE 1: The Media Server should maintain the status for both uplink and downlink traffic and a separate interface for supporting downlink and uplink is expected to be defined in this specification.

NOTE 2: WebRTC-enabled UE should support streaming functions for uplink and downlink traffic. Therefore a new entity in UE may be defined.

### 4.3.4 RTC-5: Control transport interface

The RTC-5 interface is an interface between the RTC MSH and the RTC AF. It is used to convey configuration information from the RTC AF to the RTC MSH and to request support for a starting/ongoing WebRTC session. The configuration information may consist of static information such as the following:

- Recommendations for media configurations

- Configurations of STUN and TURN server locations

- Configuration about consumption and QoE reporting

- Discovery information for WebRTC signalling and data channel servers and their capabilities

The support functionality includes the following:

- RTC MSH receives the configuration information

- RTC MSH informs the RTC AF about a WebRTC session and its state

- RTC MSH requests QoS allocation for a starting or modified session

- RTC MSH receives notification about changes to the QoS allocation for the ongoing WebRTC session

- RTC MSH receives the updated information about the WebRTC session with the RTC STUN/TURN/Signalling function, e.g. to identify a WebRTC session and associate it with a QoS template

The RTC functionality that offer application functions to the WebRTC application may equally be provided by Application Servers (RTC AS) instead of RTC AF. These then use a dedicated interface RTC-3 to request configurations and network support for the ongoing WebRTC sessions from the RTC AF.

### 4.3.5 RTC-6: Client API

The RTC MSH is a function in the UE that provides access to RTC support functions to the native WebRTC applications. These functions may be offered on request, i.e., through the RTC-6 interface, or transparently without direct involvement of the application. The RTC MSH may assist indirectly in the ICE negotiation by providing a list of STUN and TURN server candidates that offer RTC functionality. The RTC MSH also collects QoE metric reports and submits consumption reports. It may also offer media configuration recommendations to the application through RTC-6.

### 4.3.6 RTC-7: Client interface

This is an interface between RTC Access Function and the native WebRTC Application to directly communicate media-specific information.

### 4.3.7 RTC-8: Application interface

This is a proprietary interface between the application and the application provider, which may be used to exchange configuration information related to the RTC session or the application.

### 4.3.8 RTC-10: RTC AS to another RTC AS interface

This is an interface to exchange the media stream and metadata between one instance of the Media AS and another for the purpose of distributed service chaining over multiple RTC ASs. This may be optionally present when each RTC Client has different Media Function in the separately-located RTC AS.

### 4.3.9 RTC-11: UE configuration interface

The RTC-11 is an interface between the RTC MSH and the RTC Access Function, both in the RTC Client, to configure media session handling and/or media access. It may not be exposed as an API to application developers but may be in form of a direct communication. The RTC Access Functionhides away all details of the QoS allocation and network support from the application. It autonomously and transparently invokes the functions offered by the RTC MSH to provide support for the RTC session.

### 4.3.10 RTC-12:

This interface is used to exchange the WebRTC traffic between different RTC Access Functions in different UEs. When RTC-12 is used for the purpose of peer-to-peer media transport, it shall be a subset of RTC-4m.

Next change

### 4.4.2 Extended RTC architecture for Edge Computing

#### 4.4.2.1 General

The RTC architecture can be extended to add support for media processing in the edge. The extended architecture is an integration of the RTC architecture defined in TS 26.506 with the architecture for enabling Edge Applications defined in TS 23.558 [7] and TS 26.501 [6].

The extended RTC architecture supports both client-driven as well as Application Function-driven management of the edge processing session.

The RTC Application Provider may request the deployment of edge resources as part of the Provisioning Session.

- The RTC Application Provider provisions the edge provisioning through RTC-1, a similar fashion as defined in TS 26.512 clause 7.10, enabling client-driven and/or Application Function driven edge configuration.

- In the client-driven approach, the WebRTC Application becomes aware of the support of edge processing in the network and takes steps, such as using the EDGE-5 APIs, to discover and locate a suitable RTC AS instance in the Edge DN, similar to the process defined in TS 26.501 clause 8.1.

- In the Application Function driven approach, the RTC Application Provider requests RTC AF to deploy edge processing for the media sessions of the corresponding Provisioning Session, similar to the process defined in TS 26.501 clause 8.2. The WebRTC Application may get aware of the deployed EAS through the Application Service Provider through RTC-8 or through the RTC MSH through RTC-5 (and possibly RTC-6). The EAS is provided together such that the associated can be made by UE between two set of data. Additionally, the EAS may also be discovered through other means, such as DNS resolution with support from the DNS server (e.g., EASDF/DNS resolver) as specified in 3GPP TS 23.548 .

When the WebRTC application is a web application, the implementation of the EDGE-5 interface to discover the RTC AS/EAS location by accessing the EEC is difficult as the Web browser providers may not implement interfaces necessary for supporting edge enabled RTC applications/services. Also, in the Application Function-driven approach the Application Client (AC) and EEC are not used to discover the RTC AS/EAS location.

To resolve the above EAS discovery issue in the Application Function-driven approach and when the WebRTC application is a web application, the EAS information can be shared with the RTC MSH by the RTC AF using RTC-5 interface.

NOTE: Other methods that can be used for sharing EAS information (e.g., sharing EAS hostname to the WebRTC application by RTC-8 or by other means and then using DNS resolution) are FFS.



Figure 4.4.2-1: Edge-enabled RTC architecture

NOTE: This architecture diagram is an example for collaboration scenario 2. For the collaboration scenario 3 and 4, other subfunctions in RTC AS, as specified in clause 4.2, should be further represented.

Next change

## 5.4 Call flow for Network-supported RTC sessions (CS#2)

The MNO offers access to trusted ICE functionality to UEs that wish to participate in RTC sessions. The session establishment takes into account the configured trusted ICE functions.

The call flow is as follows.



Figure 5.4-1: Call flow for Network-supported RTC sessions (collaboration scenario 2)

The working assumptions are:

- The application on UE1 and remote endpoint use an external WebRTC signalling function to establish the WebRTC session.

0. A provisioning session may have been created by the AP with the MNO.

Call flow using network-supported RTC session is achieved through the following steps:

1. The RTC AF uses the RTC-5 interface to provide the RTC MSH with a list of trusted STUN/TURN servers that the UE may use for establishing RTC sessions.

2. The application queries the RTC MSH for the list of trusted ICE servers.

3. The UE discovers and tests the ICE candidates to validate that they are suitable for the connection.

4. The application on UE1 and the remote endpoint use an external RTC signalling function to exchange information about ICE candidates and to exchange the SDP offer/answer.

Then, the WebRTC session is established using the most suitable ICE candidate.

5. The STUN or TURN server in ICE function, upon reception of the allocation request by the application (or RTC Access Function) may extract the 5-Tuple information for each of the media sessions and convey the information to the Network Support AF in RTC AF for requesting QoS assistance.

6. The Network Support AF uses the N5 interface to request QoS allocation. It may request differential charging based on pre-existing provisioning for these sessions.

7. Confirmation of QoS allocation is notified to the Network Support AF and the RTC MSH.

8. The Network Support AF will also subscribe to events related to the QoS flows of the WebRTC session with the PCF and SMF.

9. The Network Support AF receives notifications about any changes to the QoS flows of the WebRTC session from the PCF or the SMF. Then, the Network Support AF sends notifications to the ICE function (STUN/TURN server).

10. The STUN/TURN server may forward the bitrate recommendation to the RTC MSH, if the allocation session is still active.

11. Alternatively, the MSH may interact with the UE Modem to trigger to query the recommended bitrate on the uplink or downlink direction.

12. The UE Modem then sends the ANBRQ (Access Network Bit Rate Query) signalling to the RAN as defined in TS 38.321 [8] for NR access and TS 36.321 [9] for LTE access.

13. The RAN, based on the network status, returns the recommended bitrate to the UE modem as requested. The recommended bit rate is in kbps at the physical layer at the time when the decision is made.

NOTE 1: The UE may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified. The UE may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified.

NOTE 2: The eNodeB may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified.

NOTE 3: The recommended/queried bitrate as signalled over the LTE and NR access is defined to be in kbps at the physical layer. The uplink/downlink bitrate at the physical layer is $r\_{UL/DL}=\frac{\sum\_{k}^{}L\_{k}}{T}$,where$L\_{k}$is the bit-length of the *k*-th successfully transmitted/received TB by the UE within the window *T*. In TS 36.321[9] and 38.321[8], a window length of 2000 ms is applied.

14. The application may act on the bitrate recommendation, e.g. by reducing the uplink media bitrate.

15. Media traffic is delivered to remote endpoint. If TURN server is present in the configuration, RTC-4m interface is involved.

## 5.5 Call flow for MNO-Facilitated RTC sessions (CS#3)

In collaboration scenario 3, MNO hosts the WebRTC sessions by providing a trusted WebRTC signalling function in the RTC AS. In addition, a trusted media server is also present in RTC AS to support SFU and MCU functionality. Note that, when the WebRTC application is a web-based application, the RTC MSH function is not supported.

The call flows for this scenario when RTC MSH is involved are as shown in Figure 5.5.1.

 

Figure 5.5.1: Call flows for MNO facilitated RTC sessions (collaboration scenario 3)

The RTC Application Provider may create a ***Provisioning Session*** with the RTC AF and starts provisioning the usage of the RTC Streaming session between two endpoints. During the establishment phase, the used features such as content consumption measurement, logging, collection and reporting; QoE metrics measurement, logging, collection and reporting; dynamic policy; network assistance; are negotiated and detailed configurations are exchanged.

The RTC Application Provider ***Provisioning*** ***Session*** phase is optionally performed prior to the establishment of any related WebRTC sessions by the RTC Application Provider. Detailed procedure is addressed in clause 5.2.1.

The ***ICE candidate discovery*** phase is performed with the following steps in an MNO-facilitated RTC system:

1. Configuration information: The RTC AF uses the RTC-5 interface to provide the RTC MSH with a list of trusted STUN/TURN servers, trusted WebRTC signalling function and data channel servers and their capabilities. The UE may use this configuration information for establishing RTC sessions.

2. ICE Servers request: The application queries the RTC MSH for the list of trusted ICE servers.

3. ICE candidate validation: The UE discovers and tests the ICE candidates to validate that they are suitable for the connection.

The ***WebRTC session establishment*** phase is performed with the following steps in an MNO-facilitated RTC system:

4. Query configuration information: The RTC Access Function queries the RTC MSH for the WebRTC signalling function information. In some cases where the signalling is not handled by RTC Access Function, the native WebRTC application queries the RTC MSH for the WebRTC Signalling server information.

5. Configuration information: RTC MSH sends the WebRTC signalling function and data channel servers and their capabilities information to RTC Access Functionor in some cases with native WebRTC application.

In ***SDP exchange*** phase, two or more RTC Clients exchange signalling information related to the WebRTC session such as ICE candidates, SDP offer/answer using the trusted WebRTC signalling function.

NOTE: Figure 5.5.1 illustrates that SDP offer is generated by the RTC Access Function or native WebRTC Application. However, in SFU/MCU mode, SDP offer is generated by Media Function in RTC AS.

6. SDP offer: The RTC Access Function or native WebRTC Application creates a request with SDP offer which includes the ICE candidates and sends it to the WebRTC signalling function.

7. Determine remote endpoint location: The WebRTC signalling function uses the registration information to locate the remote endpoint

8. SDP offer: The WebRTC signalling function forwards the request to remote endpoint.

9. SDP answer: Upon accepting the offer, remote endpoint responds to signalling function with SDP answer.

10. SDP answer: WebRTC signalling function sends the SDP answer to the UE1.

With this, a WebRTC session is established between RTC Clients using the most suitable ICE candidate and the WebRTC signalling function.

The ***Dynamic policy*** phase is then performed to allocate QoS for the media streams of the RTC session with the following steps:

11. QoS request: The WebRTC signalling function sends a request to RTC AF for the allocation of QoS for the session. The RTC AF sends a request to the PCF to allocate QoS for the media streams of the RTC session

12. Confirmation: PCF or SMF confirms the successful allocation of network support or QoS allocation.

If the Network support function feature is supported in the RTC AF, then the Network Support Function AF (NS-AF) offers the bitrate recommendation request API based on existing policy templates, through the usage of either the Npcf\_PolicyAuthorization API over N5 interface, or the Nnef\_AFSessionWithQoS over N33 interface to the PCF. If no corresponding AF application session context already exists, the NS-AF uses the Npcf\_PolicyAuthorization\_Create method over N5 interface with the appropriate service information to create and provision an application session context. The ***Network assistance*** phase is performed with the following steps in an MNO-facilitated RTC system.

13. Subscribe to QoS events: The NS-AF also subscribes to events related to the QoS flows of the WebRTC session with the PCF and SMF.

14. QoS events: The NS-AF receives notifications about any changes to the QoS flows of the WebRTC session from the PCF or the SMF.

15. QoS notifications/ Bitrate recommendations: The NS-AF may send notifications to the RTC MSH about the changes in QoS flow. When the allocated session is active, the RTC MSH forwards the bitrate recommendation to the application.

16. Adjust media bitrate: The WebRTC application may act on adjusting the bitrate recommendation, e.g., by reducing the uplink media bitrate.

After successful creation of a WebRTC session and the bitrate negotiations, the actual ***WebRTC session*** over 5G may start:

17. Media transfer:

a) The WebRTC Application may connect to the selected TURN server and/or Media Function in the RTC AS through the RTC-4m interface and real-time communication starts, and the media is delivered to the remote endpoint.

b) In some cases, a peer-to-peer connection is also possible and the media is delivered directly to the remote endpoint.

18. Method calls and notifications: Supporting information about the WebRTC session is passed from the RTC Access Function to the RTC MSH.

19. Reporting, network assistance, and dynamic policy: The RTC MSH exchanges supporting information about the WebRTC session with the RTC AF.

20. End session: The WebRTC Application informs the RTC Access Function that the RTC session has ended. It is also sent to the WebRTC Signalling Function to terminate the session.

21. Session ending event: The RTC Access Function informs the RTC MSH about the end of the RTC session.

22. Final reporting: The RTC MSH performs any final reporting to the RTC AF.

Next change

## 6.1 Client-driven Management of RTC Edge Processing

The detailed call flow for client-driven management of edge processing session is shown in Figure 6.1-1.



Figure 6.1-1. Client-driven management of RTC edge processing

The ***Edge Computing Provisioning*** phase is a provisioning phase, that may be repeated several times (e.g., to extend edge processing coverage to new geographical areas or to increase the capacity of an already provisioned area). All steps in this phase are optional and performed on need basis. The steps are:

1. Spawn ECS: In this step, a new ECS instance is instantiated to manage new or increased demand for edge processing.

2. Spawn RTC AF: In this step, a new RTC AF that is edge-enabled is instantiated to handle new or increased demand for WebRTC sessions with edge processing.

3. EES Configuration: The EES is configured for a specific Edge Data Network (EDN).

4. EES Registration with ECS: The EES registers with the ECS that is in authority over the target EDN.

The ***RTC Application Provider Provisioning*** phase is performed prior to the establishment of any related WebRTC sessions by the RTC Application Provider. Subsequent updates to the provisioning session are possible.

5. Create Provisioning Session: In this step, the RTC Application Provider creates a new provisioning session.

6. Provision RTC features: In this step, the RTC Application Provider may create different configurations such as QoS support, charging, collection of consumption, offering STUN/TURN servers, WebRTC signalling function, Edge Processing, etc.

The WebRTC Application initiates a new RTC session:

7. Application Initialization: The user launches the WebRTC Application. The WebRTC application performs any required initialization steps.

8. Start session: The WebRTC Application invokes the RTC Access Function with appropriate real-time streaming access parameters.

9. Session starting event: The application informs the RTC MSH about the start of a new WebRTC session over 5G.

10. Retrieve service access information: The RTC MSH retrieves Service Access Information from the RTC AF appropriate to the WebRTC session.

11. Determine eligibility for requesting edge resources: Using information from the Service Access Information, the RTC MSH determines whether the WebRTC session is eligible for requesting edge resources.

If the eligibility criteria are met in the previous step, the UE discovers an EAS instance offering RTC AS functionality in the ***Client-based Edge Computing Discovery*** phase:

12. Locate EAS instances: The RTC MSH asks the EEC to discover the location of one or more suitable EAS instances offering the RTC AS capability that can serve the application.

13. Locate local EES: The EEC queries the ECS for a suitable EES (EDGE-4 API).

14. Register with EES: The EEC registers with the selected EES (EDGE-1 API).

15. Request list of “RTC AS” EAS instances: The EEC queries the EES for one or more EAS instances offering the “RTC AS” capability that can serve the session, using EAS discovery filters (see Table 8.5.3.2-2 in [2]) provided by the Application Client, e.g. “RTC AS” for EAS type, appropriate values for service feature(s), and other EAS characteristics.

The optional sub-flow ***RTC AS Provisioning*** is for provisioning an additional RTC AS instance if a suitable EAS instance offering the **"**RTC AS**"** capability cannot be located. The steps are:

16. Check resource template: The RTC AF checks the provisioned edge processing resource template for the related application to determine the requirements of the application.

17. Instantiate new EAS/RTC AS: The RTC AF requests the MnS to instantiate a new RTC AS EAS instance with the specified requirements and considering parameters provided in the query by the EEC.

18. Spawn RTC AS instance: The MnS creates a new instance of the EAS offering RTC AS capability with the requested placement and resources.

19. EAS configuration: The newly instantiated RTC AS EAS instance is configured.

20. Register EAS with EES: The newly instantiated EAS instance registers itself with the triggering EES.

21. Configure provisioned features: This may include configuring and launching the server-side application in the RTC AS.

Completion of UE Edge Computing Discovery phase:

22. List of suitable “-RTC AS” EAS instances: The EES/RTC AF responds to the EEC with a list of “RTC AS” EAS instances and their characteristics in an EAS discovery response (see Table 8.5.3.3-1 in [16]).

23. Select preferred “RTC AS” EAS instance: The AC and/or EEC select(s) a “RTC AS” EAS instance from the provided list, based on the AC’s desired criteria.

After successful discovery of a “RTC AS” EAS instance, the actual ***WebRTC session*** over 5G may start:

24. Media transfer: The WebRTC Application connects to the selected EAS “RTC AS” and the real-time streaming starts.

25. Method calls and notifications: Supporting information about the WebRTC session is passed from the RTC Access Function to the RTC MSH.

26. Reporting, network assistance, and dynamic policy: The RTC MSH exchanges supporting information about the WebRTC session with the RTC AF.

27. End session: The WebRTC Application informs the RTC Access Function that the RTC session has ended.

28. Session ending event: The RTC Access Function informs the RTC MSH about the end of the RTC session.

29. Final reporting: The RTC MSH performs any final reporting to the RTC AF.

## 6.2 AF-driven Management of RTC Edge Processing

The detailed call flow for AF-driven management of edge processing session by using the RTC MSH is shown in Figure 6.2-1.



Figure 6.2-1. AF-driven management of RTC edge processing

The steps are:

1. Steps 1-4 as described in TS 26.501 clause 8.1.

2. Create Provisioning Session: In this step, the RTC Application Provider creates a new provisioning session.

3. Provision RTC features: In this step, the RTC Application Provider may create different configurations such as QoS support, charging, collection of consumption, offering STUN/TURN servers, WebRTC signalling function, edge processing, etc.

4. RTC AS provisioning if need, as described in Figure 6.1-1, steps 16-21.

The WebRTC Application initiates a new RTC session:

5. Start session: The WebRTC Application invokes the RTC Access Function with appropriate real-time streaming access parameters.

6. Session starting event: The application informs the RTC MSH about the start of a new WebRTC session over 5G.

7. Retrieve Service Access Information: The RTC MSH retrieves Service Access Information from the RTC AF appropriate to the WebRTC session.

8. Determine eligibility for requesting edge resources: Using information from the Service Access Information, the RTC MSH determines whether the WebRTC session is eligible for requesting edge resources.

9. Start the media streaming as defined in Figure 6.1-1, steps 24-26.

10. Continue the final steps as defined in Figure 6.1-1, steps 27-29.

End of changes