ETSI TS 126 238 V18.0.0 (2024-05)

Universal Mobile Telecommunications System (UMTS);
LTE;
5G;
Uplink streaming
(3GPP TS 26.238 version 18.0.0 Release 18)

Reference
RTS/TSGS-0426238vi00

Keywords
5G,LTE,UMTS

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

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 https://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.

Introduction

The Framework for Live Uplink Streaming (FLUS) is an enabler for live media streaming from a source entity to a sink entity. FLUS offers an IMS-based and a non-IMS-based instantiation. The IMS/MTSI-based instantiation enables the establishment of live media streaming between two UEs or between a source entity and a sink entity, within and across operator networks. Compared with MTSI, where limited types of QoS for speech or video media are used, FLUS can provide a wider range of QoS operation, e.g., in the maximum delay, available bandwidth or target packet loss rate.

In the non-IMS-based instantiation, it is possible to operate FLUS as a more generic framework that is controlled through a RESTful API and that supports other media plane protocols (i.e. not based on IMS or MTSI).

In addition to providing a wider range of QoS operation over radio links, other advanced functionalities of FLUS, such as the signalling of immersive media, can be used to complement existing 3GPP services.

1 Scope

The present document defines a FLUS source entity and a FLUS sink entity that can support point-to-point transmission of speech/audio, video, and text. It defines media handling (e.g., signalling, transport, packet-loss handling, and adaptation). The goal is to ensure a reliable and interoperable service with a predictable media quality while allowing for flexibility in the service offerings.

A FLUS source entity, which may be embedded in a single UE, or distributed among a UE and separate audio-visual capture devices, may support all or a subset of the features specified in this document.

When used as a generic framework, only the F-C procedures for establishing the FLUS session are required to be supported by the FLUS source and the FLUS sink entities, and no other feature or procedure specified in this document is mandated. Impact on the service quality and network capacity is left to the discretion of the implementation and the service utilizing the framework. For example, configuration of media formats and codecs follows the requirements of the respective service.

When offered as part of a 3GPP IMS/MTSI service, the FLUS source and the FLUS sink entities are required to support the IMS control plane and media plane procedures, and the service quality is determined by the MTSI service policy.

The specification is written in a forward-compatible way in order to allow additions of media components and functionality in releases beyond Release 15.

.

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] Recommendation ITU-R BS.2051-1 (06/2017): "Advanced Sound System for Programme Production".

[3] ISO/IEC 23090-2:2019: "Information technology - Coded representation of immersive media - Part 2: Omnidirectional media format".

[4] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[5] 3GPP TS 26.235: "Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs".

[6] 3GPP TS 24.147: "Conferencing using the IM Multimedia (IM) Core Network (CN) subsystem; Stage 3".

[7] 3GPP TS 23.003: "Numbering, addressing and identification".

[8] 3GPP TS 33.203: "3G security; Access security for IP-based services".

[9] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[10] 3GPP TS 33.328: "IP Multimedia Subsystem (IMS) media plane security".

[11] IETF RFC 7231: "Hypertext transfer protocol (HTTP/1.1): Semantics and Content" (June 2014).

[12] 3GPP TS 24.229: "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[13] 3GPP TS 23.401: "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access".

[14] 3GPP TS 23.682: " Architecture enhancements to facilitate communications with packet data networks and applications".

[16] 3GPP TS 26.501: "5G Media Streaming (5GMS); General description and architecture".

[17] ISO/IEC DIS 23090-8: "Information technology -- Coded representation of immersive media -- Part 8: Network based media processing" (under development).

[18] IETF RFC 4574: "The Session Description Protocol (SDP) Label Attribute" (August 2006).

[19] 3GPP TS 36.321: "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification".

[20] 3GPP TS 38.321,: "NR; Medium Access Control (MAC) protocol specification".

[21] OMA-ERELD-DM-V1_2-20070209-A: "Enabler Release Definition for OMA Device Management, Approved Version 1.2".

[22] 3GPP TR 26.939: " Guidelines on the Framework for Live Uplink Streaming (FLUS)".

[23] 3GPP TS 26.511: "5G Media Streaming (5GMS); Profiles, codecs and formats".

[24] IETF RFC 7798 (2016): "RTP Payload Format for High Efficiency Video Coding (HEVC)"

[25] 3GPP TS 26.445: "Codec for Enhanced Voice Services (EVS); Detailed Algorithmic Description".

[26] IETF RFC 4867 (2007): "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs".

[27] IETF RFC 6416 (2011): "RTP Payload Format for MPEG-4 Audio/Visual Streams".

[28] OASIS Standard, MQTT Version 5.0, https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[29] 3GPP TS 24.147, Conferencing using the IP Multimedia (IM); Core Network (CN) subsystem;

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions 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].

FLUS session: A logical association between a FLUS source and a FLUS sink within which media content can be sent from the source to the sink.

Media session: A subset or part of a FLUS session between a FLUS source and a FLUS sink including the duration to establish the Media Media session, the time period during which media content can be sent from the FLUS source to the FLUS sink and the duration to terminate the MMedia session. One or more MMedia sessions are delivered during a FLUS session. A MMedia session may be established and controlled by a well-defined control protocol.

Media stream: The content sent from a FLUS source to a FLUS sink within a MMedia session.

3.2 Symbols

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

<symbol> <Explanation>

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].

FLUS	Framework for Live Uplink Streaming
HMD	Head Mounted Display
IMS	IP Multimedia Subsystem
MO	Management Object
NBMP	Network Based Media Processing

4 System architecture

4.1 General

This clause introduces the system architecture for FLUS.

4.2 System

4.2.1 General

Figure 4.2-1 depicts the architecture showing the relevant entities for providing an uplink streaming service.

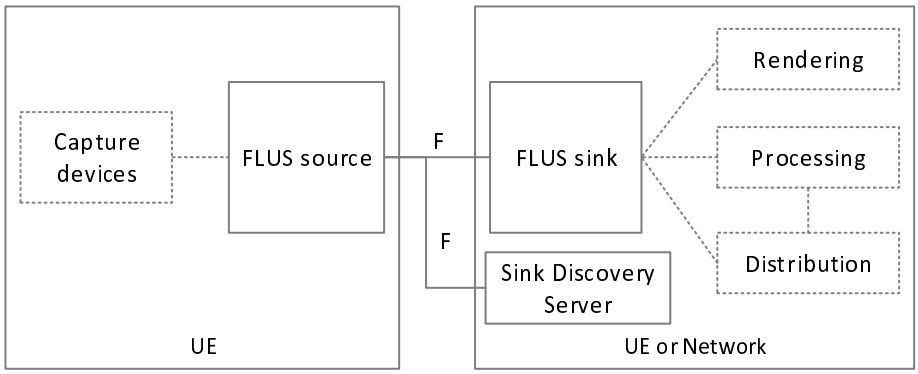


Figure 4.2-1: FLUS architecture

The uplink streaming service architecture is based on a FLUS source located in a UE and a FLUS sink located in either another UE or the network.

The FLUS source receives media content from one or more capture devices. In the context of this specification, the capture devices are considered as parts of a UE or are connected to it.

When the FLUS sink is located in a UE, the FLUS sink shall forward media content to a decoding and rendering function.

When the FLUS sink is located in the network,

the FLUS sink may forward media content to a processing or distribution sub-function. The processing and distribution sub-functions are not in scope of the present specification. The FLUS sink may act as a Media Gateway Function (MGW) and/or an Application Function (AF).

- an optional Sink Discovery Server offers functionality that allows to discover available FLUS sinks.
- each FLUS sink provides functionality to query its capabilities and FLUS sessions.

The F reference point connects a FLUS source to a FLUS sink and optionally to the Sink Discovery Server. The F reference point enables the discovery of available FLUS sinks. In addition,. the F reference point enables the establishment and control of a single FLUS session between the FLUS source and FLUS sink.

The F reference point also enables the FLUS sink and the FLUS source to mutually authenticate and authorize each other.

FLUS source and FLUS sink are each split into:

- Media Source and Sink (F-U end-points),
- Control Source and Sink (F-C end-points),
- Remote Controller and Remote Control Target (F-RC end-points) and

The UE, the FLUS source and the FLUS sink are considered to be logical functions. Functions are not required to be located in the same physical device; they can be spread over multiple physical devices and interconnected via other interfaces.

Multiple F-RC end-points can exist in a single FLUS source. F-RC end-points are independent of a FLUS sink and depend on the offered service.

The F reference point shall support security functionality for confidentiality protection for all subfunctions.

Details of the functions are shown in Figure 4.2-2.

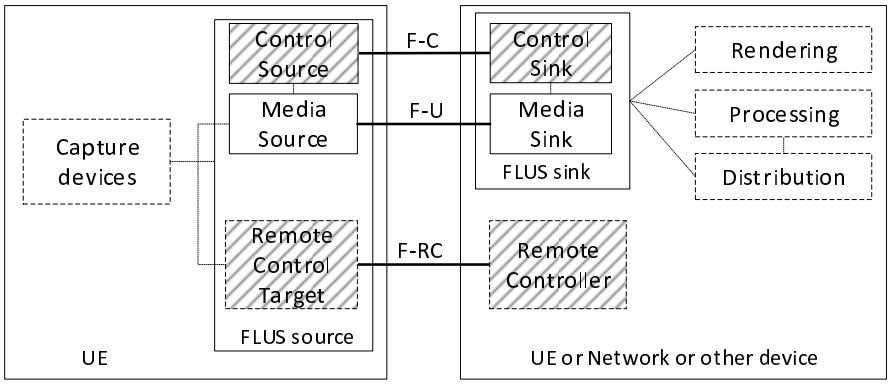


Figure 4.2-2: Sub-functions of FLUS

Note that in the above diagram, hatched-filled boxes represent FLUS control plane functionality, solid line boxes represent mandatory functionality, and dashed-line boxes corresponds to optional functionality. Also, note that F-C and F-U denote FLUS control and FLUS user plane interactions, respectively, and do not represent reference points.

Control Source and Sink (F-C): FLUS control plane functionality including the associated processing by the FLUS sink of the uploaded media for subsequent downstream distribution, plus FLUS media instantiation selection. F-C may also support configuration of static metadata for the Media session.

Remote Controller and Remote Control Target (F-RC): The Remote Control Target on the FLUS source receives control messages. The messages affect the behaviour of the Media Source in the FLUS source. Examples of commands issued to the Remote Control Target are start or stop of the media upstreaming process in the FLUS source.

Media Source and Sink (F-U): FLUS user plane functionality which includes setup of one or more Media sessions and subsequent media data transmission via Media streams. In some cases, a Media session establishment protocol (e.g. IMS session set-up for MTSI-based FLUS instantiation) is necessary.

NOTE: F-C is not needed when the FLUS sink is an MTSI client that is only capable of rendering. In such event, logical control plane functions such as media and session descriptions and support for a FLUS session establishment are encapsulated in FLUS user plane functionality.

F-C represents the interactions associated with the creation and modification of the configuration of the FLUS sink. F-C allows the Control Source to:

- select a FLUS media instantiation,

- provision static metadata associated with each Media session present in the FLUS session,

- discover, select and configure the processing and distribution sub-functions.

In addition to the delivery of control messages, F-RC represents the interactions associated with the discovery, creation, modification and selection of the FLUS sink configuration by the Remote Controller. F-RC allows the Remote Controller to:

- provide the Media Sink information to the FLUS source,

- select a FLUS media instantiation,

- determine capture device settings and other FLUS source parameters.

The FLUS media instantiation is defined as part of the FLUS session. The user plane (F-U) may also contain the Media stream establishment procedures when needed. Multiple Media streams may be established for one FLUS session.

A Media stream may contain media components of a single content type, e.g., audio, or media components of different content types, e.g., audio and video. A FLUS session may be composed of more than one Media stream containing the same content type, e.g., multiple Media streams of video.

4.2.2 Uplink streaming for MTSI

The architecture of uplink streaming for MTSI is depicted in Figure 4.2-3.

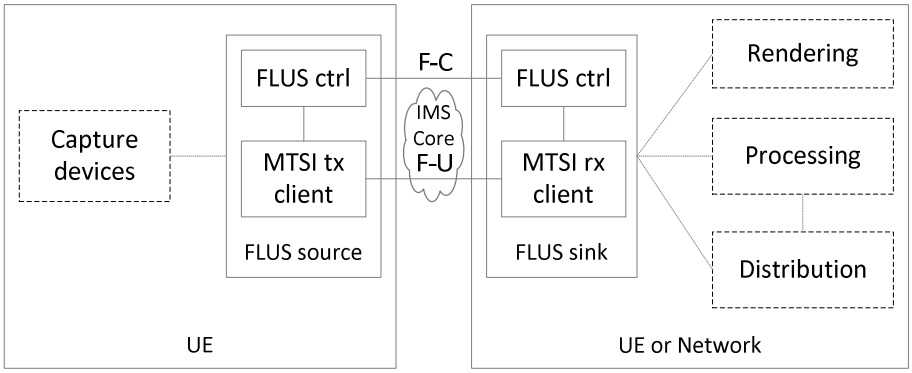


Figure 4.2-3: Uplink streaming for MTSI

The reception function of an MTSI client, i.e., the "MTSI rx client" as shown in Figure 4.2-3, is used to realize the Media Sink in the FLUS sink. The FLUS sink may be instantiated as another UE or as an MTSI rx client function in the network.

F-U contains all MTSI-related signalling.

The transmission function of an MTSI client, MTSI tx client, is used to realize the Media Source in the FLUS source.

4.2.3 Uplink streaming for PSS-based distribution

Figure 4.2-4a shows the most important interfaces for Evolved Packet System (EPS) and 5G System (5GS) for Live Uplink Streaming Services.

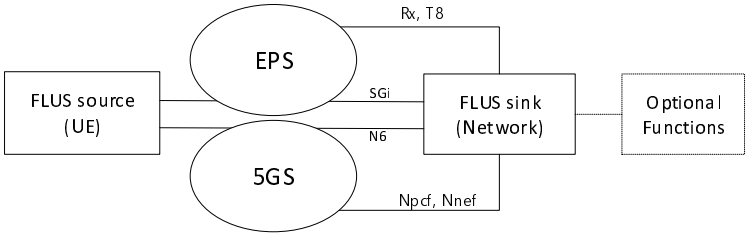


Figure 4.2-4a: Architecture for Live Uplink Streaming

An uplink streaming service requires at least a FLUS source and a FLUS sink. A FLUS sink is located at the SGi reference point in case of EPS [13] and at the N6 reference point in case of 5GS [15]. When QoS support is needed, the FLUS sink may request QoS support using Rx [13] or T8 [14] reference points in case or EPS and using Nnef or Npcf [15] reference points in case of 5GS.

The architecture of uplink streaming for subsequent PSS distribution is depicted in Figure 4.2-4b.

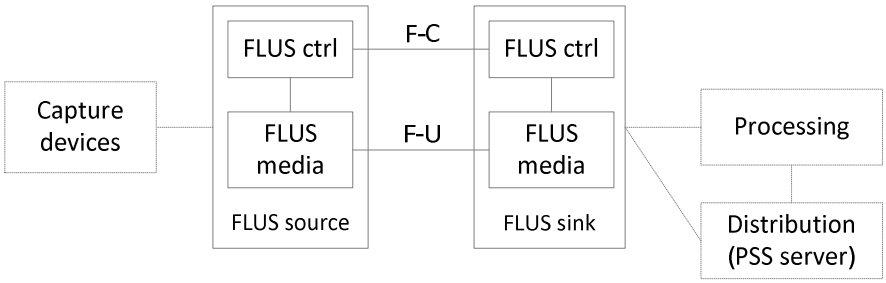
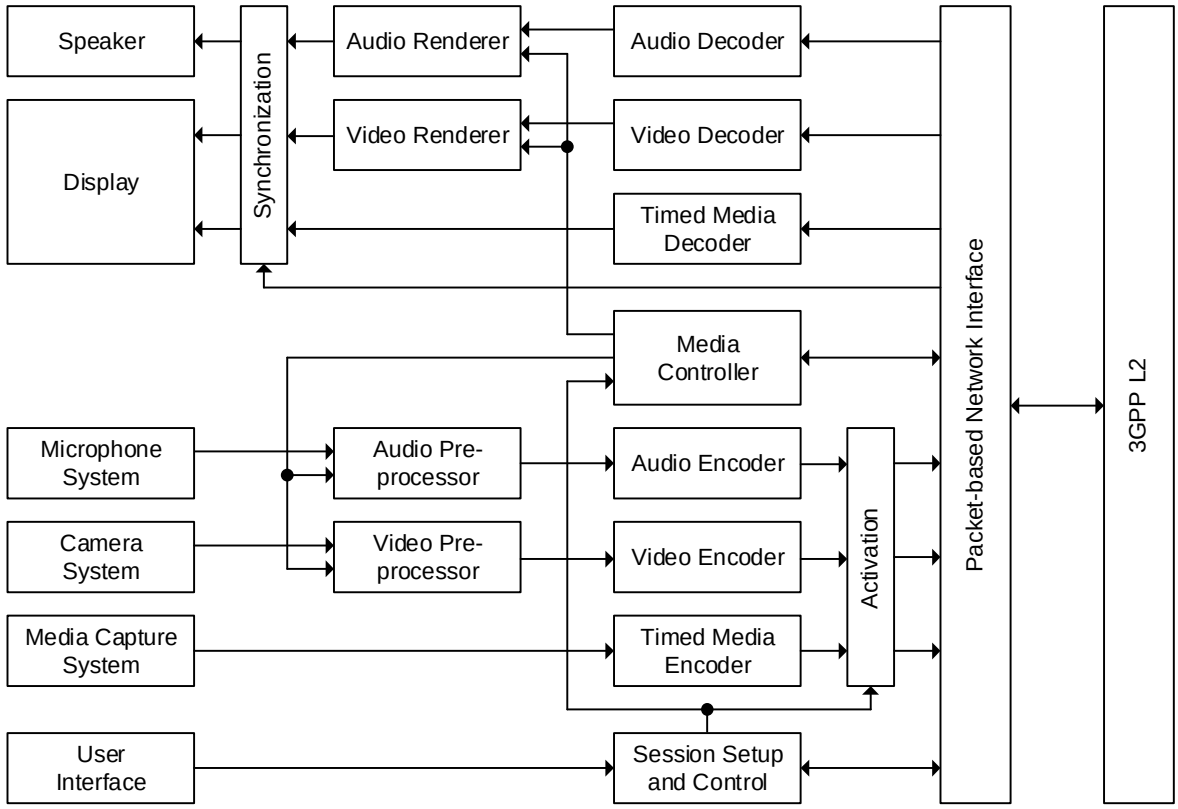


Figure 4.2-4b: Uplink streaming for PSS

The PSS Content Source is located on the UE side and contains the FLUS source.

4.3 Terminal

The functional components of a terminal including both a FLUS source and a FLUS sink that uses 3GPP access are shown in figure 4.3-1.



NOTE 1: It is not required that all components of a FLUS source or a FLUS sink are included in a terminal.

Figure 4.3-1: Functional components of a FLUS source and a FLUS sink

The scope of the present document is to specify media handling and interaction, which includes media control, as well as transport of media and control data.

As indicated in figure 4.3-1, whether or not to employ or how to realize the audio/video renderer, media controller, or audio/video pre-processor is left to the configuration of a FLUS source and a FLUS sink, and their implementation. For these functional components, this document only defines the signalling of the media controller, which will be used to define the handling of media.

Timed media may include text, graphics, etc.

If a terminal including a FLUS source uses 3GPP access only for uplink media transmission, its functional components are shown in figure 4.3-2.

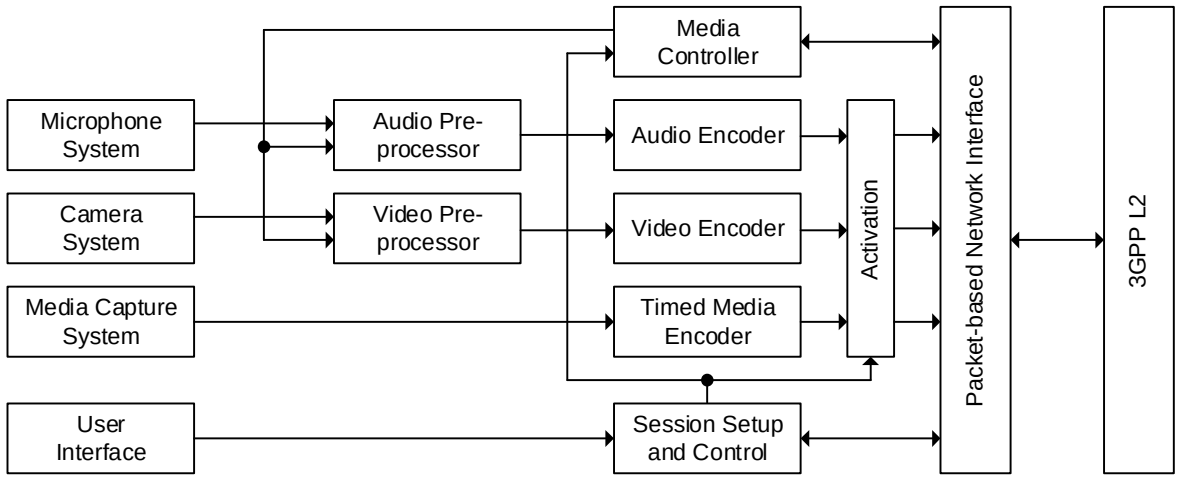


Figure 4.3-2: Functional components of a FLUS source for uplink media transmission

4.4 Procedures

4.4.1 General

A FLUS session may include one or more media streams. Media streams are time-bound to the FLUS session to which they belong to. When a media stream is active, the FLUS source can send media content to the FLUS sink.

The establishment and / or teardown of a Media session can be remotely controlled / influenced using remote control and / or assistance messaging.

Figure 4.4-1 depicts an example relationship of a FLUS session with a single Media session containing two media streams. The FLUS Session is established once the FLUS source and the FLUS sink contain matching configurations.

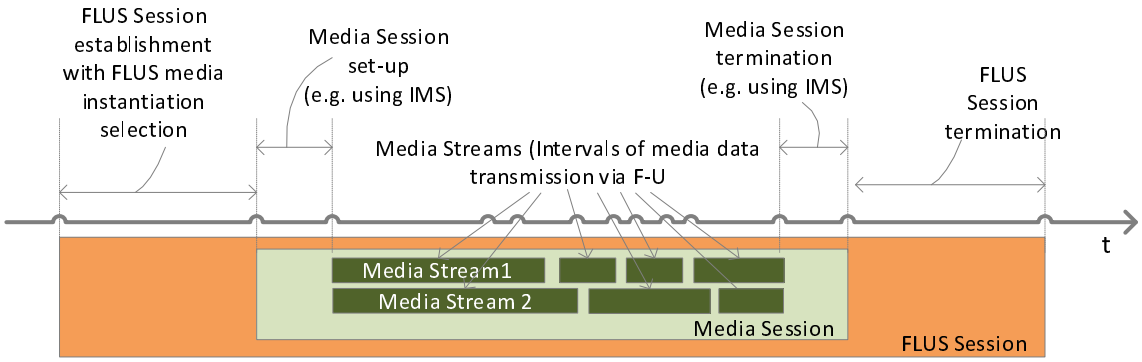


Figure 4.4-1: FLUS, Media sessions and Media streams

When the FLUS sink is located in a UE and the UE renders the received media content directly, the FLUS session may be implicitly present, e.g. it may be realized through IMS/MTSI.

When the FLUS sink is located on the network side and provides Media Gateway functionality, the FLUS session is used to select the Media session instantiation and to configure any processing and distribution related sub-functions.

The MPEG NBMP [17] Workflow Description Document may be used to describe the media processing tasks at FLUS sink to be performed on received media components from the FLUS source. See Annex B for further details.

4.4.2 FLUS session establishment

It is assumed that the FLUS source has the necessary information to establish an F-C connection to a FLUS sink, e.g., in terms of a SIP-URI or an HTTP URL.

4.4.3 FLUS session update

The FLUS session establishment procedure creates a FLUS session resource, which is then configured through the FLUS session update procedure, for example, resulting in the selection of the Media session instantiation.

The FLUS session update procedure includes the following FLUS session configuration parameters:

- Selection of the Media session instantiation
- Provision of session specific metadata,
- Setting a description of the processing of the media data that the FLUS sink is to perform,
- Configuration of the distribution and storage options for the media data

4.4.4 FLUS sink capability discovery

This procedure allows a FLUS source to discover capabilities of a FLUS sink.

The FLUS sink capabilities include

- Processing capabilities such as
  - Supported input formats, codecs and codec profiles / levels, resolutions, frame rates
  - Transcoding with formats, output codecs, codec profiles / levels, bitrates, etc,
  - Reformatting with output format,
  - Combination of input Media streams, e.g. network based stitching, mixing,
  - Recognition or synthesis of media
- Distribution capabilities such as
  - Storage capabilities,
  - CDN Origin capabilities and CDN Origin Server base URLs,
  - Forwarding, including supported forwarding protocol and associated security procedures

4.4.5 FLUS session termination

The FLUS source may explicitly terminate a FLUS session and all its provisioned and active Media sessions. Alternatively, the FLUS session is automatically terminated, when the last Media session of the FLUS session is terminated.

4.5 FLUS source systems

4.5.1 Introduction

FLUS supports different source systems. A source system description contains metadata on the captured media sources and their inter-relationships and is made available to the FLUS sink such that the FLUS sink can properly interpret and process the received Media streams. FLUS can be deployed with source systems as defined in this specification, or with proprietary source systems. A source system is identified by a unique identifier. That identifier and the corresponding source system description are provided to the FLUS sink during the FLUS session establishment and/or update procedures. The interactions between the FLUS source and FLUS sink, in terms of HTTP-based RESTful API request/response mechanisms, are specified in clause 7.

A set of 3GPP-defined source systems is specified in this clause.

4.5.2 General source systems

A generic source system transmits a collection of Media streams that can be combined by the FLUS sink to create a full media experience. FLUS provides the ability to stream continuous media data from a FLUS source to a FLUS sink.

Media streams within one source system are time-synchronized and self-descriptive in the source system. Media streams are identified by a unique identifier in a source system.

Each Media stream shall be self-declarative in the announced source system.

Media streams may be of the following types: video, audio, text or metadata.

A source system may declare additional metadata in the namespace of the source system.

The FLUS source may query for a media processing function to fulfil the required action through standardized interfaces, e.g., [17].

Table 4.5-1: Generic FLUS source system description

Source System			Description	IMS-based Instantiation SDP Parameters	Non-IMS-based Instantiation Parameters
	Source System Identifier		A URI that uniquely identifies the source system	a=3gpp-flus-system:<urn>	JSON Object (SourceSystemIdentifier)
	Configuration		Provides source system-specific configuration parameters for the source system.	a=3gpp-flus-configuration:<base 64 encoded>	Not applicable
	Media Stream		Descriptions of the Media streams that are defined by the source system	m= (one per stream)	Defined by the F-U instantiation
		Identifier	A unique identifier shall be associated with the each Media stream. If not present, then the source system description should contain sufficient information to uniquely identify the Media stream in the source system.	a=label:flus… (RTP SSRC being the unique identifier, may be implicit if a=ssrc is not included)	Defined by the F-U instantiation (may be implicit)
		Stream Configuration	Describes the details of this Media stream, including the encoding, the metadata, etc. The Media stream is self-describing in the source system. The definition of a single Media stream, in the context of session description information, is provided during the FLUS session establishment, e.g. by a media line in SDP.	a=3gpp-flus-media-configuration:<base 64 encoded>	Defined by the F-U instantiation
		Bandwidth Requirement	Indication of the required network bandwidth to transport this Media	b=AS:<bw> a=bw-info:	Defined by the F-U instantiation

				streams of the source system. Mandatory if QoS is required		
		QoS Requirement		Indication of the required network QoS parameters to transport this Media streams of the source system. Recommended if specific loss and/or latency are required	a=3gpp-qos-hint:	Defined by the F-U instantiation
		Transmission Direction		Optional indication of the transmission direction of this Media streams	a=sendonly (mandatory)	Defined by the F-U instantiation
		Codec		Codec Identifier	a=rtpmap: (mandatory)	Defined by F-U instantiation.
		Codec configuration		Codec-specific configuration information for this Media streams in the source system	a=fmtp:	Defined by F-U instantiation
			Media Type	Media type of each Media stream: audio, video, text, etc...	m=<type> (mandatory)	Defined by F-U instantiation
			Media Transport and Control	Identification of the media transport protocol, operating over IP, for this Media stream in the source system	m=<type> <port> RTP/AVP (mandatory) RTP/AVPF (if video rate adaptation is desired)	Defined by F-U instantiation.

4.5.3 Vendor-specific source system

A vendor-specific source system shall be identified by a unique vendor-specific urn identifier such as `urn:[com]:[vendor_x]:[system_y]`

4.5.4 Default source system

The default source system shall be identified by the unique identifier `urn:org:3gpp:flus:default:2017`. This source system is used for planar media formats.

4.5.5 3DOF FLUS source system

4.5.5.1 Introduction

The three-degrees of freedom (3DOF) source system can be defined in a way that enables the user to view the media content sent by the source system from a single observation point in 3D space.

A 3DOF source system is identified with the unique identifier `urn:org:3gpp:flus:3dof:2017`.

The coordinate system is defined, as well as the basic structure to describe this source system.

4.5.5.2 Coordinate system

The direction in which a FLUS sink is interested can be represented using a coordinate system shown in figure 4.5-1, which consists of three axes, X, Y, and Z [3]. The X axis is equal to back-to-front axis, Y axis is equal to side-to-side (or lateral) axis, and Z axis is equal to vertical (or up) axis. Three angles, pitch, yaw, and roll, are measured around the X, Y, and Z axes respectively. As the torso is assumed to be fixed, the angles are assumed to be generated by the movement of the head. If seen from the origin to the positive direction of each axis, indicated by the arrows, each angle increases clockwise. The X-Z plane is aligned with the ground. When looking in the positive direction of Z axis, all angles are zero. The value ranges of yaw and roll are both −180.0, inclusive, to 180.0, exclusive, degrees. The value range of pitch is −90.0 to 90.0, inclusive, degrees.

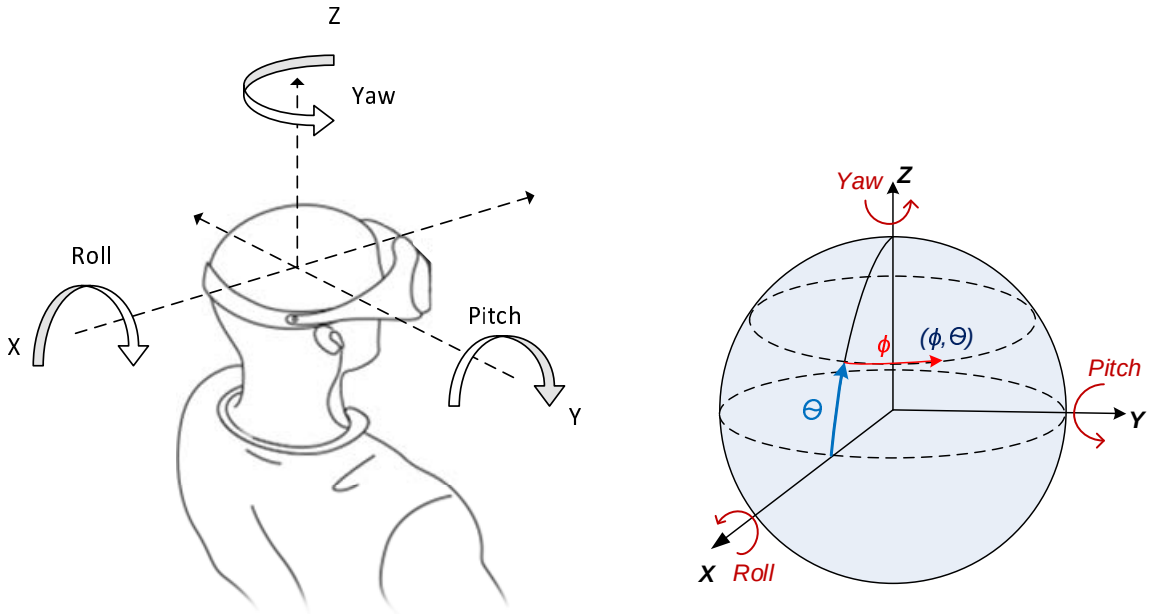


Figure 4.5-1: Coordinate system at media receiver

A source system is defined as a capturing device that has a center point from which the content is captured.

The captured content is represented in a reference system that enables a consumer of FLUS captured content to be able to look around from a single observation point in 3D space defined by the position of central capturing device(s).

This ability to look around and listen from a center point in 3D space is defined as 3 degrees of freedom (3DOF).

- Tilting side to side on the X-axis referred to as *Roll*ing
- Tilting forward and backward on the Y-axis, referred to as *Pitch*ing

- Turning left and right on the Z-axis, referred to as *Yaw*ing

It is worth noting that this center point is not necessarily static - it may be moving. The center point is defined by a physical location.

The content may be combined with simultaneously captured audio that may be 3D.

Additional timed media sources may be captured and streamed to the FLUS sink.

Signals may represent a 2D signal or a spherical signal. Spherical signals defined in this specification are represented in a spherical coordinate space in angular coordinates (ϕ, θ) for use in omnidirectional video applications and 3D audio. The viewing and listening perspective is from the origin sensing/looking/hearing outward toward the inside of the sphere. The spherical coordinates are defined so that ϕ is the azimuth (longitude, increasing eastward) and θ is the elevation (latitude, increasing northward) as depicted in figure 4.5-1.

Depending on the applications or implementations, not all angles may be necessary or available at the FLUS source. The 360 video and the 3D audio may have a restricted coverage. For example, it may not be possible for a FLUS source to control media encoding parameters related to roll or pitch, e.g., when a wheel-shaped camera with a series of lenses is located on a flat surface, in which case only yaw needs to be fed back. The means to measure these angles is outside the scope of this document.

For video, such a center point may exist for each eye, referred to as stereo signal. And obviously, the video consists of three color components, typically expressed by the luminance (Y) and two chrominance components (U and V).

Although this coordinate system is defined for FLUS sink, and a FLUS source may use a typical spherical coordinate system as defined in [2], where the layout of loud speakers is represented with azimuth and elevation, the FLUS source should take appropriate actions based on the geometric information it receives from the far-end. If such a spherical coordinate system is used, the direction at the FLUS source, when both azimuth and elevation are zero, corresponds to the direction at FLUS sink when pitch, yaw, and roll are all zero.

4.5.5.3 Descriptive Parameters

In order to describe the media and metadata components of the system, the description language needs to provide information on the source system as well as each media component. The 3DOF FLUS source system is described in Table 4.5-2.

Table 4.5-2: 3DOF FLUS source System Description

3DOF FLUS source system			Cardinality	Description
	Source System Identifier		URI	Identifier of the source system as a URI, and in this specification shall be set to: `urn:org:3gpp:flus:3dof:2017`
	Description			Detailed description of the source system including static metadata, etc. , and which shall be conveyed in a separate namespace from that of the Source System Identifier.
	Video Stream		0 … 1	At most one video stream is contained. If present, the Media stream is identified by the media type "video", such that no additional identifier is needed. The video stream shall contain sufficient information to be self-declarative in the coordinate system as defined in clause 4.5.5.2.
		Identifier	0 … 1	Unique identifier of the video stream in the source system. If not present, the source system should contain sufficient information to uniquely identify the video stream in the source system.
		Type	1	Provides the type of the contained video stream - 360 mono: The video represents a single spherical video.

				- 360 stereo: The video contains two spherical videos, one for each eye. - Unknown: will be provided by source system.
		Description	1	Describes the details of the video stream including the encoding, the metadata, etc. The Media stream is self-describing in the source system.
	Audio Stream		0 … 1	If present, the Media stream is identified by the media type "audio", such that no additional identifier is needed. The audio stream shall contain sufficient information to be self-describing in the coordinate system as defined in clause 4.5.5.2.
		Identifier	0 … 1	Unique identifier of the audio stream in the source system. If not present, the source system should contain sufficient information to uniquely identify the audio stream in the source system.
		Type	1	Defines the type of the contained audio stream: - 2D: the audio describes a planar version of the surrounding audio. No heights are included. - 3D: the audio includes 3 dimensions with heights. - Unknown: will be provided by source system.
		Description	1	Describes the details of the audio stream including the encoding, the metadata, etc. The Media stream is self-describing in the source system
	Other Media Stream		0 … N	If present, each Media stream shall include an identifier that provides an exact description of each (non- video or audio) Media stream. Note that metadata is also considered as a separate Media stream. Each such Media stream shall contain sufficient information to be self-describing in the coordinate system defined in clause 4.5.5.2.
		Identifier	1	Unique identifier (for each instance) of the Media stream in the source system. The identifier value shall be provided to describe the metadata. Examples for metadata description are 4CC codes in the file format.
		Description	1	Describes the details of this Media stream including the encoding, the metadata, etc. The Media stream is self-describing in the source system

4.5.6 Media production FLUS source system

In the media production use case for FLUS, typically it is the media production centre that controls the FLUS source devices that deliver Media streams for the media production process. The media production centre typically is located "behind" the FLUS sink, on the network side and not shown in the system architecture. The FLUS sink acts as a Media Gateway Function. The Remote Controller function acts as proxy of the media production centre for the control of the FLUS source devices.

When a FLUS source is set up and prepared for an event, it proceeds, either by manual trigger or autonomously, to register with the Remote Controller function and the media production centre. Once a FLUS source is known to the media production center, the media production centre can take full control of that device and the FLUS sessions. This includes the initiation, adaptation and termination of FLUS sessions.

4.6 Uplink Assistance

4.6.1 Uplink Assistance using UNA mechanisms

The Uplink Network Assistance (UNA) feature enables a FLUS source to improve the QoE of content up-streaming sessions, and is provided by the Network Assistance Server (NAssS). The NAssS is out-of-band with respect to the media up-stream path. The UNA communication is independent from the FLUS media (F-U) and FLUS Control (F-C) communication paths, hence the UNA communication occurs in a separate path to the transfer of the Media stream. The FLUS sink does not need to be aware of the UNA function.

UNA is based on the model of the FLUS source requesting network assistance and the NAssS responding to the request. The UNA functionality may be granted to a client supporting the provision of FLUS media content with either only the first or with both of the two functions below, in both cases based on the FLUS source having made a request to the NAssS for Uplink Network Assistance:

- The NAssS indicates to the FLUS source the highest suitable media rate for the next period of up-streamed content, based on the versions that are able to be provided;
- The NAssS indicates to the FLUS source a temporary delivery boost facility for occasions when the content upstream output buffer on the client risks suffering from over-run.

Once an Uplink Network Assistance session is active, the client may issue an Uplink Network Assistance call prior to sending the next portion of upstream content to the FLUS sink. The Uplink Network Assistance call consists of a single logical signalling exchange. This exchange with the NAssS activates either the first of the above functions or a sequence of both functions; the second only if the FLUS source was granted access to the delivery boost function. If the client does not need a delivery boost, then the NAssS omits the second function in the response to the FLUS source.

The Uplink Network Assistance feature is specified as a general feature of the 5G Media streaming system [16].

4.6.2 Uplink Assistance using RAN Signaling Mechanism

FLUS uplink assistance using RAN signaling employs bitrate recommendation and request for bit rate increase (or boost) procedures, between the UE/FLUS source and the RAN (eNB in E-UTRA and gNB in NR). It is modelled after and leverages the access network bitrate recommendation mechanisms as defined in TS 26.114 [4], TS 36.321 [19] and TS 38.321 [20].

Uplink bitrate recommendation information is sent from the RAN to the UE via the logical ANBR (Access Network Bitrate Recommendation) message. A query for updated bitrate recommendation information, towards assisting UE determination of boost support, is sent from the UE to the RAN via the logical ANBRQ (ANBR Query) message. The ANBR and ANBRQ logical messages are mapped to the *Recommended bit rate MAC CE* and *Recommended bit rate query MAC CE*, respectively, as defined in TS 36.321 [19] and TS 38.321 [20]. Details on the semantics and usage of these messages for RAN signaling based uplink assistance are provided in clause 5.4.3.

Figures 4.6.2-1 illustrate the system architecture and interactions between functional entities in the UE and the network for RAN signaling based uplink assistance in the MTSI and non-MTSI based FLUS systems.

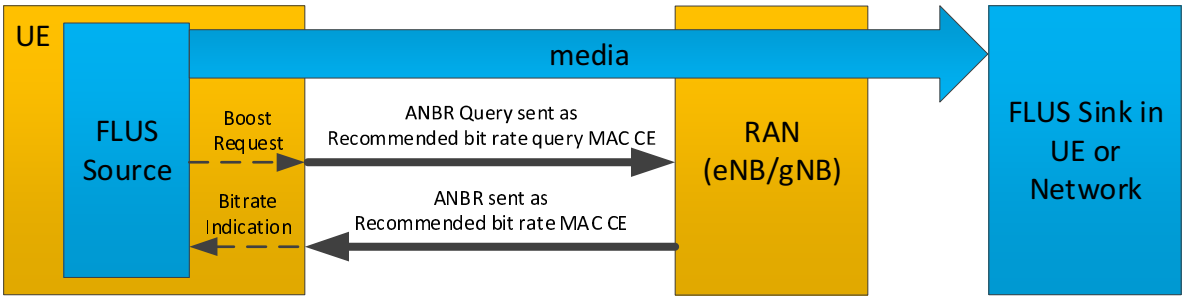


Figure 4.6.2-1 RAN signaling based uplink assistance in the FLUS system

It should be noted that in the MTSI-based FLUS system, ANBR support by the UE is indicated by the media-level SDP attribute "anbr".

5 Protocols

5.1 General

Protocol instantiations that can be configured by the functionalities in clause 7 may be used in the context of this specification.

5.2 IMS-based system

5.2.1 System configuration

5.2.1.1 Introduction

In an IMS-based FLUS system, the FLUS control plane or F-C enables the configuration and selection of a FLUS sink for the reception of Media streams delivered over the FLUS user plane or F-U during a FLUS session.

5.2.1.2 FLUS sink configuration and selection

The means for discovery or configuration, and selection by the FLUS source of the FLUS sink depend on the deployment scenario and location of the FLUS sink, as described in sub-clauses 5.2.1.2.1 and 5.2.1.2.2.

5.2.1.2.1 UE-based FLUS sink

In this deployment scenario, the FLUS sink is a directly-targeted recipient of the media content streamed by the FLUS source, The FLUS source is expected to be aware of the identity of the FLUS sink, in the form of a SIP URI, which will be included in the Request-Line of the SIP INVITE message. There is no need for a separately-defined FLUS sink configuration and selection procedure in the delivery of the SIP INVITE from the FLUS source to the FLUS sink.

5.2.1.2.2 Network-based FLUS sink

In this deployment scenario, the FLUS sink is a network ingest server that performs post-processing of the Media streams(s) uploaded from the FLUS source, for subsequent distribution to the recipient UEs. It acts as a SIP Application Server (AS) in the IMS network architecture, and its identity shall be in the form of a "FLUS Factory URI.

The FLUS source may be pre-provisioned with the FLUS Factory URI of the FLUS sink. In that case, the FLUS source shall send the INVITE message with the Request-Line-URI set to the value of that URI. If the FLUS Factory URI is not pre-provisioned in the UE, the UE shall construct a default FLUS Factory URI (similar to the procedures defined in clause 13.10 of TS 23.003 [7] regarding default Conference Factory URI construction) in either of the following formats:

a) sip:flus@flus-factory.operator.com, when the UE of the FLUS source contains the ISIM application and assumes that the name of its home network domain is 'operator.com', or

b) sip:flus@flus-factory.ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org, when the UE of the FLUS source does not contain the ISIM application, and has a home network domain name of 'ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org'.

Such a FLUS Factory URI shall in turn be used by the S-CSCF to select an appropriate FLUS sink (AS) to which the INVITE message from the FLUS source is routed for IMS session establishment.

5.2.1.3 FLUS management object

The UE may be pre-provisioned with a FLUS sink URI through a 3GPP FLUS Management Object.

The FLUS Management Object shall have a Management Object Identifier: "urn:oma:mo:ext-3gpp-flus:1.0". The MO shall be compatible with OMA Device Management protocol specification version 1.2 and above as described in [21]. The following nodes for FLUS configuration are defined:

Node: /<*X*>

This interior node specifies the unique object id of a FLUS Management Object. The purpose of this interior node is to group together the parameters of a single object.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

The following interior nodes shall be contained if the FLUS sink in the terminal supports the FLUS Management Object.

/<X>/Sink/<X>

This node is a collection of information about a FLUS sink

- Occurrence: OneOrMore
- Format: node
- Minimum Access Types: Get

/<X>/Sink/<X>/SIP_URI

This leaf node provides the SIP URI for the FLUS sink that is described by the parent node.

- Occurrence: One
- Format: string
- Minimum Access Types: Get

/<X>/Sink/<X>/Capabilities

This leaf node provides a URL to a JSON or XML document that describes the capabilities of the FLUS sink. The format of the document is outside the scope of this specification.

- Occurrence: ZeroOrOne
- Format: string
- Minimum Access Types: Get

The structure of this FLUS MO is depicted in Figure 5.2-1.

/<*X*>/Sink/<*X*>/Ext

The Ext is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne

- Format: node
- Minimum Access Types: Get

/<*X*>/Ext

The Ext is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

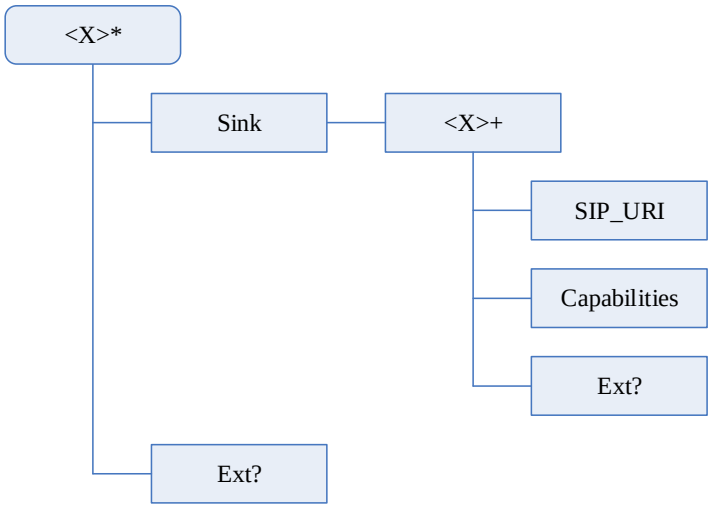


Figure 5.2-1: FLUS management object tree

5.2.2 Session management

The IMS-based FLUS System uses the SIP protocol for all session management. The FLUS sink and FLUS source shall support the IMS procedures as defined by TS 24.229 [12].

Use of the "a=3gpp-qos-hint:" SDP attribute, as described by TS 26.114 [4] clause 6.2.7.4, is recommended for FLUS media descriptions if specific QoS in the form of maximum loss and/or latency is required.

In addition, the following SDP offer/answer procedures apply for "a=label:flus…" (see also [18]):

- Generating the SDP offer:
  - "a=label:flus…" shall be included in every media description used for FLUS media in the SDP offer, where the string after "a=label:" shall start with the characters "flus" but may be followed by more characters, indicated below through an ellipsis ("…").
- Generating the SDP answer:
  - "a=label:flus…" shall be kept in every media description in the SDP answer where "a=label:flus…" was received in the SDP offer and that is accepted by the SDP answerer to be used for FLUS media. The label characters after the initial "flus" may differ between offer and answer.
  - If "a=label:flus…" is *not* included in a media description in the received SDP offer, "a=label:flus" shall not be included in the corresponding media description in the SDP answer.
- Receiving the SDP answer:

- If an accepted SDP media description in the received SDP answer contains "a=label:flus…", and if "a=label:flus…" was also included in the corresponding media description in the SDP offer, the media description shall be considered as negotiated for use with FLUS media. The label characters after the initial "flus" shall not be included in the comparison between SDP offer and SDP answer.

- If an accepted SDP media description in the received SDP answer does *not* contain "a=label:flus", and if "a=label:flus" was included in the corresponding media description in the SDP offer, the SDP answerer likely does not know of FLUS media and the SDP offerer should send an SDP re-offer with this media description disabled (port set to 0).

- If an accepted SDP media description in the received SDP answer contains "a=label:flus…", and if "a=label:flus…" was *not* included in the corresponding media description in the SDP offer, the SDP answerer is using FLUS media "a=label:flus…" incorrectly and the SDP offerer shall send an SDP re-offer with this media description disabled (port set to 0).

5.2.3 Data transport

The FLUS source and sink that implement the IMS-based FLUS system shall support data transport as specified in section 7 of TS 26.114 [4].

5.3 Generic FLUS system

5.3.1 System configuration

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

5.3.2 Session management

5.3.2.1a FLUS sink capability discovery

A FLUS session is the association between a source and a sink. The Session Management procedures defined in this section target the FLUS sink configuration and management of the FLUS session.

5.3.2.1b FLUS sink discovery

This procedure allows a Control Source to discover a list of available FLUS sinks.

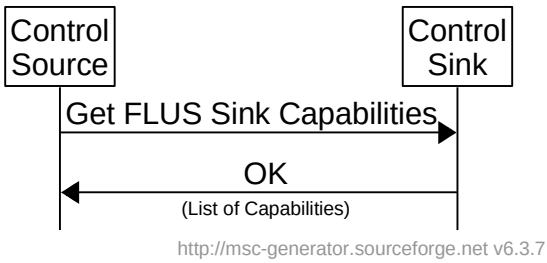


Figure 5.3.2.1b-1: Get current a list of available FLUS sinks

1. The Control Source sends the FLUS sinks request to the FLUS Sink Discovery Server.

2. The Sink Discovery Server provides a list of FLUS sinks

5.3.2.2 Create a FLUS Sink Configuration

It is assumed that the Control Source has selected a FLUS sink and acquired the necessary information (e.g. the HTTPS URL) to establish an F-C connection to the Control Sink of the FLUS sink.

The procedure allows a Control Source to create a new FLUS Sink Configuration. Configuration properties and in particular FLUS media instantiation selection is added or modified in subsequent procedures.

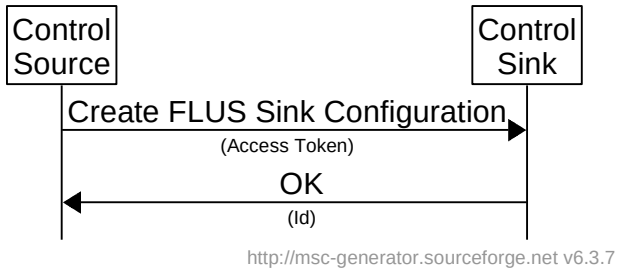


Figure 5.3.2.2-1: FLUS Sink Configuration creation

1. The FLUS Sink Configuration is created. The Control Source provides a valid access token.
2. On successful creation, the Control Sink of the FLUS sink responds with the resource id of the FLUS Sink Configuration. FLUS Sink Configuration properties are fetched and modified with subsequent transactions.

5.3.2.3 Get FLUS Sink Configuration properties

The procedure allows a Control Source to fetch the current FLUS Sink Configuration.

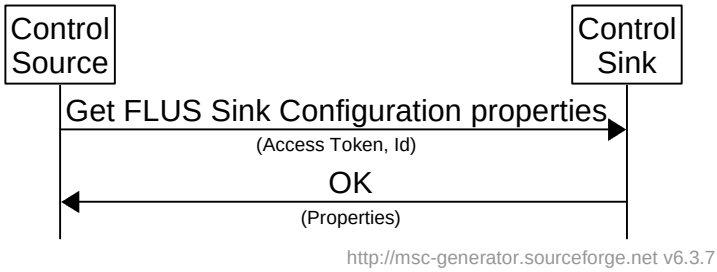


Figure 5.3.2.3-1: Get current FLUS Sink Configuration properties

1. The Control Source sends along with the FLUS Sink Configuration request, the access token and the resource id of the FLUS Sink Configuration to the Control Sink .
2. The Control Sink provides the FLUS Sink Configuration properties in response.

5.3.2.4 Update a FLUS Sink Configuration

The procedure allows a Control Source to update the current FLUS Sink Configuration.

Figure 5.3.2.4-1: FLUS Sink Configuration update

The Control Source may first fetch the current FLUS Sink Configuration using the Get FLUS Sink Configuration properties procedure.

1. The Control Source modifies the properties of the FLUS Sink Configuration resource. The procedure may allow modification of individual properties or all properties.

2. The Control Sink updates the resource identified by the id of the FLUS Sink Configuration.

5.3.2.5 FLUS session termination

The Control Source may explicitly terminate a FLUS session and all its provisioned and active Media sessions. Alternatively, the FLUS session is automatically terminated, when the last Media session of the FLUS session is terminated. The procedure allows a Control Source to terminate a FLUS session. All Media streams will be terminated automatically with the termination of the FLUS session.

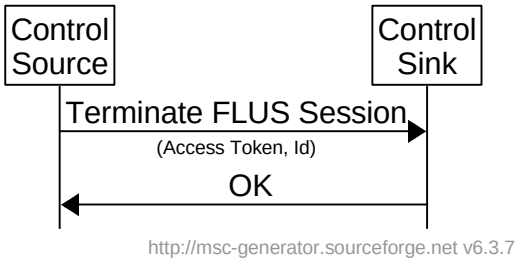


Figure 5.3.2.5-1: FLUS session termination

1. The Control Source sends the terminate FLUS session command. The access token and the resource id of the FLUS session is provided as input.

2. The Control Sink terminates the FLUS session, including all active Media streams and acknowledges the reception of the request.

5.3.2.6 Void

5.3.2.7 Void

5.3.2.8 Void

5.3.2.9 Void

5.3.2.10 Session establishment for Remote Control

When remote control is needed, the Remote Control Target in the FLUS Source is provisioned with the relevant information to establish one or more F-RC sessions. The Remote Control Target creates the F-RC sessions and then listens for incoming messages.

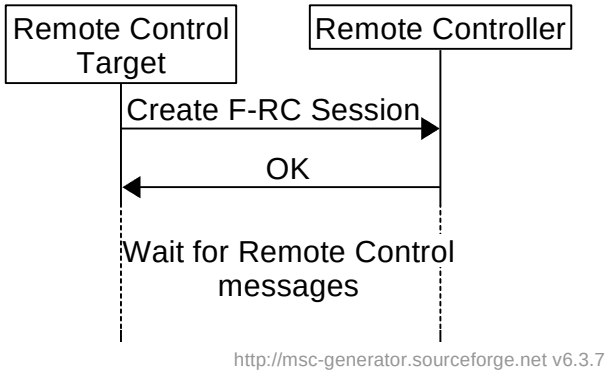


Figure 5.3.2.10-1: FLUS F-RC Session creation

1. The F-RC session is established triggered by the Remote Control Target.
2. On successful creation, the Remote Control Target receives a positive acknolwedgement.

The Remote Control Target starts waiting for remote control related messages.

5.3.2.11 FLUS Source capability discovery

This procedure allows discover capabilities of the FLUS Source.

Figure 5.3.2.11-1 shows the sequence diagram of the FLUS Source capability discovery message exchange using F-RC.

The set of FLUS Source capabilities is specified in clause 5.3.5.

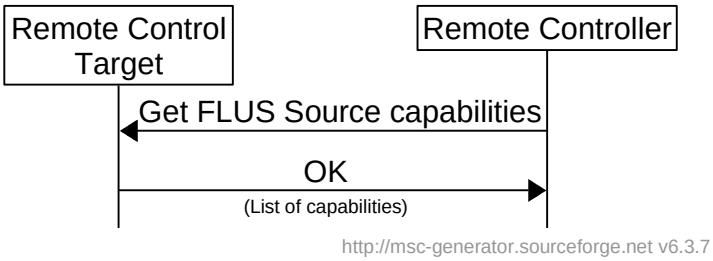


Figure 5.3.2.11-1: Get current FLUS source capabilities

1. The Remote Controller functions sends the FLUS Source capability request to the Remote Control Target of the FLUS Source.
2. The Remote Control Target provides a list of FLUS Source capabilities in response.

5.3.2.12 Remote FLUS Source configuration creation

When the FLUS Source has established a F-RC session, the Remote Controller function can create FLUS Source Configurations. FLUS Source Configuration properties and in particular FLUS media instantiation selection is added in subsequent procedures.

Figure 5.3.2.12-1 shows the sequence diagram of the remote FLUS Source Configuration establishment message exchange.

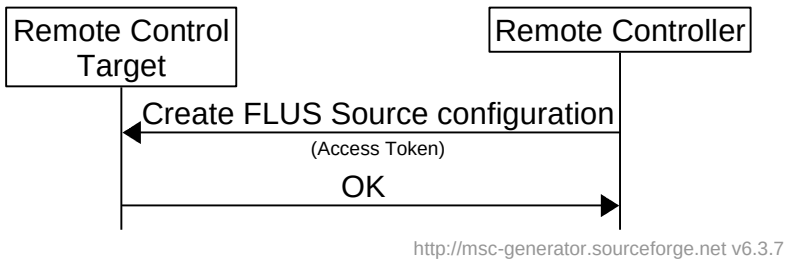


Figure 5.3.2.12-1: Remote FLUS Source configuration creation

1. The FLUS Source Configuration is created. The Remote Controller function provides a valid access token.
2. On successful creation, the Remote Control Target acknowledges the creation. FLUS Source Configuration properties (such as the FLUS Session information to identify the FLUS Sink resources) are fetched and modified with subsequent transactions.

5.3.2.13 Get FLUS Source configuration properties

This procedure allows a Remote Controller function to obtain the current FLUS Source configuration.

Figure 5.3.2.13-1 shows the sequence diagram of the FLUS Source Configuration properties retrieval message exchange.

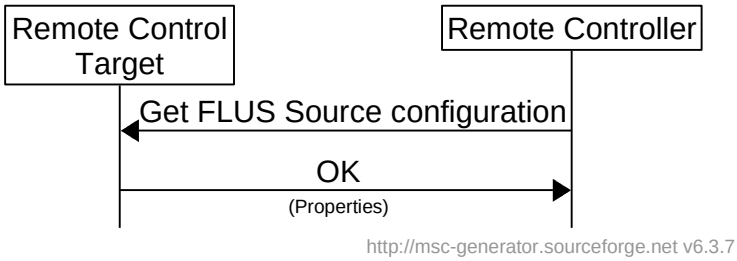


Figure 5.3.2.13-1: Get current FLUS Source Configuration properties

1. The Remote Controller function sends along with the FLUS Source Configuration properties request, the access token.
2. The Remote Control Target of the FLUS source provides the FLUS Source Configuration properties in response.

5.3.2.14 Update FLUS Source configuration

The procedure allows a Remote Controller function to update the current FLUS Source configuration.

Figure 5.3.2.14-1 shows the sequence diagram of the Remote FLUS Source Configuration update message exchange.

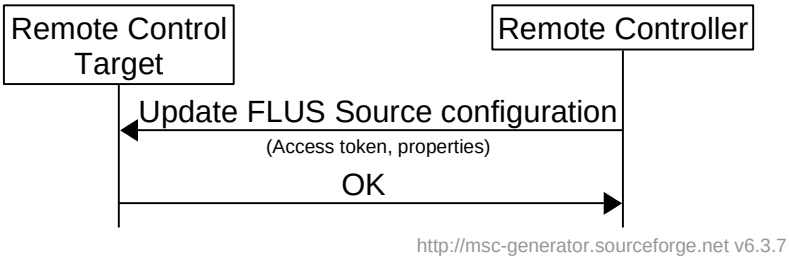


Figure 5.3.2.14-1: Remote FLUS Source Configuration update

1. The Remote Controller function modifies any or all of the properties of the FLUS Source Configuration resource.

2. The FLUS source updates the resource.

5.3.2.15 Delete a FLUS Source configuration

This procedure allows a Remote Controller function to delete a FLUS Source configuration. All active Media sessions and streams will be terminated automatically with the termination of the FLUS session.

Figure 5.3.2.15-1 shows the sequence diagram of the Remote FLUS Source configuration deletion message exchange.

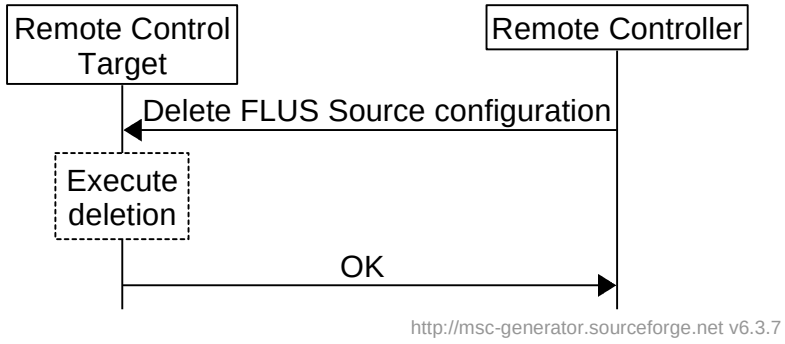


Figure 5.3.2.15-1: Remote FLUS Source configuration deletion

1. The Remote Controller function sends the Delete FLUS Source configuration command.

2. The Remote Control Target of the FLUS source deletes the FLUS Source configuration, including all active Media streams. The Remote Control Target may discard the FLUS session configuration.

3. The Remote Control Target acknowledges the execution of the request.

5.3.2.16 RAN signaling based uplink assistance

This procedure enables the RAN (eNB/gNB) to inform the UE (specifically, the MAC entity of the UE containing the FLUS source) about the recommended physical layer uplink bitrate for the associated logical channel. The bitrate recommendation sent by the RAN may be unsolicited, or as shown in the example below in Figure 5.3.2.16-1, in response (as a logical grant) to a boost request from the UE.

In Figure 5.3.2.16-1, it is assumed that the uplink streaming is performed by the FLUS source in the UE, and the sending of ANBRQ and reception of ANBR is performed by the MAC entity in the UE. Also, it should be noted that the procedures as shown in Figure 5.3.2.16-1 are applicable to either MTSI-based or non-MTSI-based FLUS instantiation.

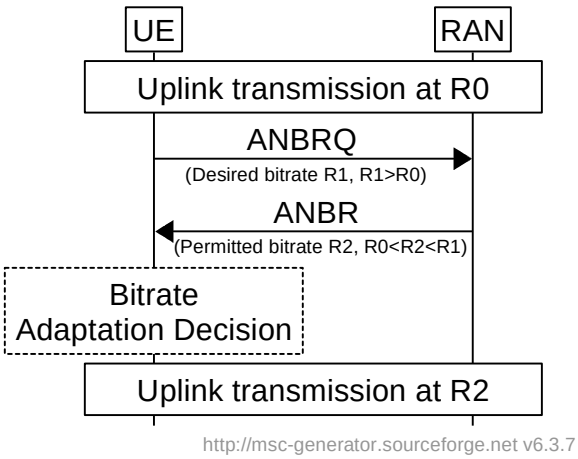


Figure 5.3.2.16-1: Uplink bitrate boost via RAN signaling

1. Initially, the UE is streaming on the uplink at bitrate R0
2. UE requests a boost on the uplink by sending ANBRQ containing the desired bitrate R1.
3. RAN grants the boost at a lower rate R2 than R1 but higher than R0, by sending ANBR containing the recommended bitrate R0 < R2 < R1.
4. UE performs bit rate adaptation decision.
5. UE streams on the uplink at the increased bitrate R2.

5.3.3 Data transport

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

5.3.4 List of FLUS sink capabilities

The FLUS sink capabilities should include the features that are listed under sub-clause 4.4.4.

5.3.5 FLUS source characterisation, capabilities and configuration properties

FLUS source devices are characterized so that the basic FLUS-related functionality can be made known to the FLUS sink.

The following FLUS source properties are specified:

- Available video format(s);
- Available audio format(s);
- Available ancillary stream format(s): subtitles / captions, content metadata – for that which is not embedded in the Media stream(s);
- Connectivity: RAN, wired – and which system/s;
- Remote control - one of: manual, remote, none – and which system/s;
- Mobility - one of: fixed, on foot, ground vehicle, airborne vehicle, surface water vehicle, underwater;
- Power – one of: Battery, battery with autonomous charging, mains.

The Remote Controller function uses the FLUS source capability discovery message exchange as specified in sub-clause 5.3.2.11 to retrieve the capabilities of the FLUS source and its attached media capture device(s).

The Remote Controller function uses the FLUS Source Configuration properties retrieval message exchange as specified in sub-clause 5.3.2.13to retrieve the current status of the FLUS source device. The status is returned in the form of the capabilities of the FLUS source populated with the applicable setting to indicate the current state of the corresponding parameter.

Some parameters can contain a particular capability setting, e.g. the current media format being transmitted. Other parameters contain one of the possible settings, possibly adding auxiliary information, e.g. when running on battery power, the current charge level and estimated operating time before needing to re-charge is indicated. When charging, the charge level and estimated time to full charge could be indicated.

In the table below, the following assertions are made:

- Table header: C stands for Create FLUS Source configuration procedure, G is for Get FLUS Source Configuration properties procedure, U is for Update FLUS Source configuration properties procedure and T is for Terminate FLUS Source configuration procedure. "I", and "O" respectively denote "request" (going **I**nto the FLUS sink), and response (going **O**ut of the FLUS Source).

- Optional ("O") means that the property may or may not be sent/received during a REST transaction. It does not necessarily mean that the property is optional. It is possible, for example, that a session is not yet active because the FLUS Source has not set the property in any previous update transaction using the PUT or PATCH HTTP method, as opposed to representing a hint on the importance of the property for the FLUS Source.

- A property marked as optional (O) in a request message may be present in the request. When not present in the request body, the property, if present in the FLUS Source, will not be updated.

- A property marked as optional (O) in a response message is only present in the response when a value is assigned or changed by the FLUS Source.

- A property marked as mandatory (M) in a response message is always present in the response. The FLUS Source provides defaults, which may be modified subsequently by the content provider.

- A blank cell in the table means "forbidden" (the property cannot be added to the request or returned by the FLUS Source, depending on the transaction direction).

Table 5.3.5-1: List of FLUS Source Configuration properties

Property Name	Property Description	CI	CO	GI	GO	UI	UO	TI
id	Identifier of the FLUS Source Configuration resource. Note that "id" is only provided within an HTTP body during the Create FLUS session response. Otherwise, "id" should be present in the message URL to identify the resource in the FLUS Source. Type: Integer; Unit: None; Default: N/A		M					
fu_instantiation	Identifier of the FLUS media instantiation that is used by this FLUS session. Vendor specific enumeration values shall start with "vnd-" followed by a unique vendor name and optionally followed by additional characters.				M	O		

	The F-U instantiation shall be provided as a globally unique URN. Type: URI; Unit: None; Default: All								
entrypoint_URL	Entry point URL information (e.g., SIP URL) for establishing the F-U connection to start the Media streaming. Details on the Entrypoint URL is F-U instantiation specific.								

5.3.6 List of FLUS Sink Configuration properties

All FLUS Sink Configuration properties, except for the resource id, are always carried in an HTTP message body. The access-token is always carried as part of HTTP headers. Except for the FLUS session creation request (where the id is not present), the resource id shall be present in the URL of all requests that relate to a specific FLUS Sink Configuration.

In the table below, the following assertions are made:

- Table header: C stands for Create FLUS session procedure, G is for Get FLUS session properties procedure, U is for Update FLUS session properties procedure and T is for Terminate FLUS session procedure. "I", and "O" respectively denote "request" (going **I**nto the FLUS sink), and response (going **O**ut of the FLUS sink).
- Optional ("O") means that the property may or may not be sent/received during a REST transaction. It does not necessarily mean that the property is optional. It is possible, for example, that a session is not yet active because the FLUS source has not set the property in any previous update transaction using the PUT or PATCH HTTP method, as opposed to representing a hint on the importance of the property for the FLUS sink.
- A property marked as optional (O) in a request message may be present in the request. When not present in the request body, the property, if present in the FLUS sink, will not be updated.
- A property marked as optional (O) in a response message is only present in the response when a value is assigned or changed by the FLUS sink.
- A property marked as mandatory (M) in a response message is always present in the response. The FLUS sink provides defaults, which may be modified subsequently by the content provider.
- A blank cell in the table means "forbidden" (the property cannot be added to the request or returned by the FLUS sink, depending on the transaction direction).

Table 5.3.6-1: List of FLUS Sink Configuration properties

Property Name	Property Description	CI	CO	GI	GO	UI	UO	TI
id	Identifier of the FLUS Sink Configuration resource. Note that "id" is only provided within an HTTP body during the Create FLUS session response. Otherwise, "id" should be present in the message URL to identify the resource in the FLUS sink. Type: Integer; Unit: None; Default: N/A		M					

<table>
<tr><td>fu_instantiation</td><td>Identifier of the FLUS media instantiation that is used by this FLUS session.

Vendor specific enumeration values shall start with "vnd-" followed by a unique vendor name and optionally followed by additional characters.

The F-U instantiation shall be provided as a globally unique URN.

<table><tr><td>Type</td><td>Unit</td><td>Default</td></tr><tr><td>URI</td><td>None</td><td>All</td></tr></table></td><td></td><td></td><td></td><td>M</td><td>O</td><td></td><td></td></tr>
<tr><td>entrypoint_URL</td><td>Entry point URL information (e.g., SIP URL) for establishing the F-U connection to start the Media streaming. Details on the Entrypoint URL is F-U instantiation specific.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>processing_description</td><td>This object provides a media processing description document that defines the post processing pipeline that the FLUS sink shall apply to received media components. The pipeline description may also set the distribution target (incl FLUS sink storage) for the media.

The Object has the following properties:

- type: the MIME type of the media processing description document

- document: the media processing document may be embedded in this element. The document may be base64 encoded depending on the MIME type.

- url: the URL to the media processing document.

- response-code: the response code to a request in the media processing document. The syntax of this property is defined by the MIME type.

The type and either the document property or the url property shall be provided.

The following formats are supported:

- The MPEG NBMP Workflow Resource, UTF-8 encoded,, as defined in [17], which describes the requested media processing and the desired distribution mechanism after the processing has been performed. The type field shall be set to "application/mpeg-nbmp-wdd+json" See Annex X on use of NBMP in FLUS.</td><td>O</td><td>O</td><td></td><td>O</td><td>O</td><td>O</td><td></td></tr>
</table>

5.4 RAN Signaling based Uplink Assistance

5.4.1 General

Overall description of RAN signaling based uplink assistance is provided in clause 4.6.2, and a sequence diagram on uplink bitrate boost is shown in clause 5.3.2.16. The following sub-clauses in this section contain detailed normative description of RAN signaling based uplink assistance.

5.4.2 Uplink bitrate recommendation and boost request

Notification from the RAN to the UE (its MAC entity) of the most current uplink bitrate recommendation information, on a per logical channel basis, is carried by the ANBR message as described in clause 4.6.2. Request from the UE (its MAC entity) to the RAN for updated recommended uplink bitrate information on a per logical channel basis, in the form of a bitrate boost query, is carried by the ANBRQ message also as described in clause 4.6.2. The ANBR and ANBRQ messages are modelled after, and are functionally equivalent to, the identically-named message pair ANBR/ANBRQ for access network bitrate recommendation/query as defined for MTSI in TS 26.114 [4].

5.4.3 ANBR/ANBRQ mapping to MAC signaling and nominal usage

ANBR and ANBRQ, in representing bitrate recommendation and boost request, are logical or conceptual messages, and similar to MTSI, usage of these logical messages in FLUS are mapped to the *Recommended bit rate MAC CE* and *Recommended bit rate query MAC CE*, respectively, as defined in TS 36.321 [19] and TS 38.321 [20].

The RAN employs the *Recommended bit rate MAC CE*, carried in a MAC PDU, to inform the UE (via the UE's MAC entity), and not necessarily only in response to UE query for such information, about the recommended bitrate for the associated logical channel on the uplink radio bearer. It is expected that an ANBR message will be sent whenever the RAN finds it reasonable or useful to notify the UE about a change in the recommended bitrate, such that the FLUS source is generally provided with up-to-date recommended bitrate information. Therefore, there is no provision for explicit bitrate recommendation request (nor need for such request) from the UE in RAN signaling based uplink assistance.

The UE, via its MAC entity, may send an ANBRQ (ANBR Query) message to query the RAN for updated recommended bitrate, i.e., ANBR information. An ANBRQ is typically sent by the UE when it wishes to seek RAN recommendation on increasing the bitrate of its uplink transmission on the logical channel identified in the request. Should the RAN determine that a higher bitrate can be supported, it is expected to transmit an updated ANBR in response.

5.4.4 Uplink assistance in the context of QoS

The radio bearer associated with ANBR and ANBRQ messaging can be either a GBR bearer/GFBR flow or non-GBR bearer/non-GFBR flow. The following descriptions apply to the various scenarios described below on MTSI or non-MTSI based FLUS instantiations in the context of GBR bearer/GFBR flow vs. non-GBR bearer/non-GFBR flow.

For either MTSI-based or non-MTSI-based FLUS system:

- In the case of GBR bearer/GFBR flow, and assuming GBR/GFBR < recommended bitrate < MBR/GFBR, the RAN may send ANBR recommending UE transmission at an uplink bitrate R1 greater than the current rate R0. This recommendation shall stand until the RAN sends another ANBR recommending UE transmission at a different bitrate.

- In the case of non-GBR bearer/non-GFBR flow, the RAN may send ANBR recommending UE transmission at an uplink bitrate R1 greater than the current rate R0. This recommendation stands until the RAN sends another ANBR recommending UE transmission at a different bitrate.

6 Terminal capabilities

6.1 General

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

7 Uplink Streaming Control Interface

7.1 General

7.1.0 Introduction

A FLUS source can connect to a FLUS sink and start live uplink streaming of content that is captured by a set of connected capture devices connected to the FLUS source. This section specifies REST API procedures for FLUS sink discovery, capability discovery, session setup, and session termination between the FLUS source and the FLUS sink.

7.1.1 Resources

7.1.1.0 General

F-C defines a set of resources managed at the FLUS sink that are controlled by the FLUS source. The set of resources are defined in the following table:

The request URI used in each HTTP request from the Control Source towards the FLUS sink shall have the structure defined in subclause 4.4.1 of 3GPP TS 29.501 [21], i.e.:

{apiRoot}/{apiName}/{apiVersion}/{apiSpecificResourceUriPart}

with the following components:

- The {apiRoot} shall be set as described in 3GPP TS 29.501 [21].
- The {apiName} shall be "flus".
- The {apiVersion} shall be "v1".
- The {apiSpecificResourceUriPart} shall be set as described in this clause.

The **apiPath** is defined as **{apiName}/{apiVersion}/{apiSpecificResourceUriPart}**.

The set of resources are defined in the following table:

Table 7.1.1-1: Resources for managing FLUS sessions at FLUS sink

Resource Name	Resource Type	Description
sinks	Collection resource	Allows discovery of available Sinks as described in clause 7.2. The resource server of a Sinks Resource is discovery server.
Capabilities	Instance resource	Represents a single FLUS sink as described in clause 7.1.1.1. The FLUS source can query the properties of that sink. However, a FLUS source cannot create, update, or terminate a FLUS sink. The creation, update, and termination of FLUS sinks are controlled exclusively by the network
session	Instance resource	Represents a single FLUS session resource as described in clause 7.1.1.2. The FLUS source can provision or modify a single session at the FLUS sink.
sessions	Collection Resource	Represents a collection of FLUS session resources.

The logical relationship between the above resources in show in Figure 7.1.1-1.

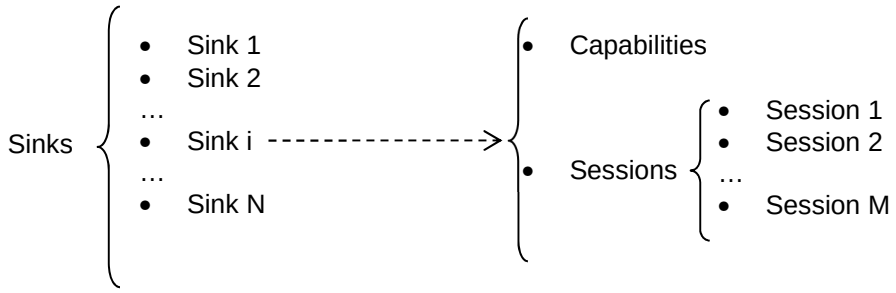


Figure 7.1.1-1: Logical relationship between FLUS resources

Note that Sinks is an array of Sink objects. Each Sink object contain a URL indicating the location of an object describing its capabilities and sessions. Therefore, the relationship between Sink object and resource showing its capabilities and sessions is logical (shown by dotted arrow in Figure 7.1.1-1).

7.1.1.1 Capabilities Resource

A capabilities resource provides a representation of the capabilities of a FLUS sink. Different properties of capabilities resources represent the capabilities of the corresponding FLUS sink.

Each capabilities resource has the set of properties described in Table 7.1.1.1-1.

Table 7.1.1.1-1: Properties of Sink Resource

Property Name	Description	Example Values
capabilities	List of supported features and instantiations by the FLUS sink. Each capability is to be expressed using an object element of an array. The object has the following attributes: - A scheme URN to identify the capability - An optional location URL, from which a description for the capability can be retrieved. The format of description is defined by the scheme URN - An optional location URL, through which the feature or instantiation can be directly accessed. The access protocols and the use of feature or instantiation are defined by the scheme.	{ "scheme" : urn:vnd:xzy:capability-name, "location" : "http://vnd.com/xzy/capability-name", "url": "http://vnd.com/xzy/capability-access" }

As described in Table 7.1.1.1-1 above, each capabilties resource describes the capabilities of the corresponding FLUS sink. A FLUS source can retrieve the capabilites resource description of a FLUS sink and make a decision if it wants to use the corresponding FLUS sink as described using the capability exchange procedure in section 7.3.

The attributes in Table 7.1.1.1-1 may be used for the following purposes:

- "scheme": to identify the capability with a unique id. This id is also used to identify the format of the capability's description document identified by the attribute "location"
- "location": the address where a document can be found that describes the detailed properties and configuration of the capability such as supported features, the configuration parameters, and the units and ranges of parameters.
- "url": the address for using the capability. For instance, if the capability needs set up, this address can be used to configure the capability. As an example, the url for a network-based media processing (NBMP) is the URL address of the NBMP Workflow Manager that enables creating, retrieving, getting an update, and deleting the workflow from this address.

7.1.1.2 Session Resource

A Session resource provides the representation of a session between a FLUS source and a FLUS sink. A FLUS source can create a session at a FLUS sink using REST API methods operating on the session resource described in this section.

Each session resource has the set of properties described in Table 5.3.6.

FLUS sources use the above session resource representation to create and manage sessions at the selected FLUS sink. Different procedures that operate on the sink and session resources are discussed in the following sections.

7.1.2 Supported Methods

The F-C API follows the RESTful design principles. All operations shall be performed using HTTP 1.1 (IETF RFC 7231 [6]) over TLS (3GPP TS 33.246 [7])

Table 7.1.2-1: Supported Methods

HTTP Method	CRUD	Resource	{apiPath}
POST	Create	Session	/flus/v1.0/sessions
GET	Read	Session	/flus/v1.0/sessions/{session-res-id}
GET	Read	Sink	/flus/v1.0/sinks
GET	Read	Capabilities	/flus/v1.0/capabilities
PUT	Replace	Session	/flus/v1.0/sessions/{session-res-id}
PATCH	Modify	Session	/flus/v1.0/sessions/{session-res-id}
DELETE	Delete	Session	/flus/v1.0/sessions/{session-res-id}

7.1.3 Error Handling

The Uplink Streaming Interface API shall use the HTTP status codes to indicate any errors that might occur in the processing of operations on FLUS resources. Unless defined otherwise, the HTTP status codes shall be interpreted as specified in IETF RFC 7231 [6]. API operations that are not successfully handled shall not leave the resource at an undefined state. The response should provide sufficient information for a human operator to understand and locate the error.

7.2 Discovery

Example: GET /flus/v1.0/sinks

The FLUS sink discovery procedure is used by the FLUS source to discover available FLUS sinks that are provided by the operator.

To query the list of available FLUS sinks, the FLUS source shall use one of the following procedures:

- The operator provides one or more FLUS sink Discovery Server URLs to the FLUS source. The Control Sources uses the pre-defined URL "apiPath" /flus/v1.0/sinks in an HTTP request to the FLUS Discovery Server URL above. The response shall be a JSON document that represents an array of object containing each the apiRoot to the entry point to the FLUS Sink as value of property "apiRoot" and optionally the capabilities of the FLUS Sinks.

- DNS discovery of the FLUS sink, using the FQDN flus.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org in the URL of form http://flus.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org/flus/v1.0/sinks/

An example json document, describing two FLUS sinks is

```
[
{"apiRoot" : "https://sink1.example.com/api", "capabilities" : ["urn:vnd:xzy:capability-name"] },
{"apiRoot" : "https://sink2.example.com"}
]
```

7.3 Capability retrieval

Example: GET /flus/v1.0/capabilities

Using this capability exchange procedure, the Control Source enquires about the supported features and instantiations of the FLUS capabilities.

To retrieve the capabilities of a FLUS sink, the FLUS source shall use the HTTP GET method on the "sink" instance resource as follows:

- the request URI with the "apiPath" is set to "/flus/v1.0/capabilities"
- the Host field is set to the FQDN of the FLUS sink

The URL in the request is set according to the information retrieved using the Discovery procedure described in section 7.2.

The capabilities resource information provides in detail the capabilities of the FLUS sink. The response from the Control Sink to the Control Source shall contain the following:

- The Content-Type field specifying the MIME type (JSON) using which the capabilitesresource information is encoded.
- The Content-Length is the length of the content body.

The above header fields are followed by the content body in the format indicated by the Content-Type field. The content body includes the detailed representation of the capabilites resource as described in section 7.1.1.1 from which the capabilities of that FLUS sink can be inferred.

The possible response messages from the Control Sink to the Control Source are shown in Table 7.3-1.

Table 7.3-1: Response status code, message, and contents for sink resource retrieval using HTTP GET

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink provides the sink capabilities to the FLUS source.
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
Note: In addition to the above response codes, the Control Source can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.4 Uplink streaming configuration

7.4.1 FLUS session properties fetch procedure

Example: GET /flus/v1.0/sessions/{session-res-id}

Sessions can be read by the FLUS source when it wishes to know the latest representation of the session resource at the FLUS sink. To fetch the properties of a FLUS session, the Control Source sends a HTTP GET request to the Control Sink as follows:

- the request URI with the "apiPath" set to "/flus/v1.0/sessions/{session-res-id}"
- the Host field is set to the FQDN of the FLUS sink

The {session-res-id} in the request URI is the session resource identifier of the session previously created between the FLUS source and the FLUS sink.

Upon receiving HTTP GET request from the Control Source, the Control Sink checks to see if such a session exists that matches the given session resource identifier. If such a session exists, the Control Sink shall respond to the Control Source with a 200 OK message along with the complete representation of the session resource. The response shall contain the following:

- The Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- The Content-Length is the length of the content body.

The content body of this response message shall be the representation of the session resource as described in sub clause 7.1.1.2

Alternatively, if such a session cannot be found by the Control Sink, it shall send a 404 Not Found message to the Control Source. If the request cannot be fulfilled, the Control Sink shall send a 403 Forbiddent message to the Control Source.

The possible response messages from the FLUS sink, depending on whether the GET request is successful or unsuccessful, are shown in Table 7.4.1-1.

Table 7.4.1-1: Response status code, message, and contents for service modification using HTTPS GET

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink provides session properties of the session resource to the FLUS source
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the Control Sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.4.2 FLUS session update procedure

The sessions at the FLUS sink can be either partially or completely modified by using the procedures as specified in clauses 7.4.2.1 and 7.4.2.2, respectively.

7.4.2.1 Partial modification of FLUS session

Example: PATCH /flus/v1.0/session/{session-res-id}

To update some of the FLUS session parameters at the sink, the Control Sourcesends an HTTP PATCH request as follows:

- the request URI with the "apiPath" is set to "/flus/v1.0/sessions/{session-res-id}"
- the Host field is set to the FQDN of the FLUS sink
- The Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- The Content-Length is the length of the content body.

The {session-res-id} in the request URI is the session resource identifier of the session whose modification is sought.

The content body of the HTTP PATCH message shall contain the updated partial representation of the session resource.

Upon receiving the HTTP PATCH request from the Control Source, the Control Sink checks to see if such a session exists at the sink. If such a sink exists, Control Sink may update the session properties based on the values from the incoming request. Upon successful update of the requested session, the Control Sink shall respond to the Control Source with a 200 OK success message indicating that the session was successfully updated. The Control Sink shall also include the session resource identifier of the session that is updated. As alternative to the 200 OK message, Control Sink may send a 204 No Content success message without any message content to the Control Source. If the session cannot be updated, the Control sink shall send a 403 message. If the session is not found, the FLUS sink shall send a 404 message.

The possible response messages from the FLUS sink, depending on whether the PATCH request is successful or unsuccessful, are shown in Table 7.4.2.1-1.

Table 7.4.2.1-1: Response status code, message, and contents for session modification using HTTP PATCH

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink provides the session resource identifier of the session that is modified
204 No Content	The request has succeeded	None
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the Control Sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.4.2.2 Full modification of FLUS session

Example: PUT /flus/v1.0/sessions/{session-res-id}

For complete update of the session parameters at the Control Source, the FLUS source sends an HTTP PUT request as follows:

- the request URI with the "apiPath" is set to "/flus/v1.0/sessions/{session-res-id}"
- the Host field is set to the FQDN of the FLUS sink
- The Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- The Content-Length is the length of the content body.

The {session-res-id} in the request URI is the session resource identifier of the session whose modification is sought.

The content body of the HTTP PUT message shall contain the updated complete representation of the session resource.

Upon receiving the HTTP PUT request from the Control Source, the Control Sink checks to see if such a session exists at the sink. If such a sink exists, Control Sink may update the session properties based on the values from the incoming request. Upon successful update of the requested session, the Control Sink shall respond to the Control Source with a 200 OK success message indicating that the session was successfully updated. The Control Sink shall also include the session resource identifier of the session that is updated. As alternative to the 200 OK message, Control Sink may send a 204 No Content success message without any message content to the Control Source. If the session cannot be updated, the Control Sink shall send a 403 message. If the session is not found, the Control Sink shall send a 404 message.

The possible response messages from the FLUS sink, depending on whether the PATCH request is successful or unsuccessful, are shown in Table 7.4.2.2-1.

Table 7.4.2.2-1: Response status code, message, and contents for session modification using HTTP PATCH

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink shall send the session resource identifier of the session that is modified
204 No Content	The request has succeeded	None
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the Control Sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.5 Session establishment

Example: POST /flus/v1.0/sessions

To create a session at the FLUS sink, the Control Source shall use the HTTP POST method on the "sessions" collection resource as follows:

- the request URI with the "apiPath" is set to "/flus/v1.0/sessions"
- the Host field is set to the FQDN of the FLUS sink
- the Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- the Content-Length is the length of the content body.

The content body of the POST request shall contain the representation of the session resource. The session resource representation includes the details such as the Media streams, their formats and codecs, the relationship between them, and the selected user plane instantiation and user plane control protocol.

Upon receipt of HTTP POST to create a session, the Control Sink shall create a session. Upon successful creation of session resource, theControl Sink shall respond back to the Control Source with a 201 success message indicating that the session is successfully created along with the session resource of the created session. The session information returned to the Control Source along with the 201 message includes the parameters applied for the session, thus confirming the parameters, and any default values assigned to session parameters which were not supplied by the Control Source in the HTTP POST message.

The session resource identifier is the identifier that uniquely identifies the session within the list of all sessions at that sink. When the Control Source receives the session resource identifier, it shall use this identifier in subsequent requests to the Control Sink to refer to this session.

Alternatively, if the creation of session is failed, the Control Sink shall send a 403 message.

The possible response messages from the FLUS sink, depending on whether the POST request is successful or unsuccessful, are shown in Table 7.5-1.

Table 7.5: Response status code, message, and contents for session creation

Status Code	Message	Contents
201 Created	Session created successfully	The FLUS sink provides the session resource identifier of the created session and detailed session information
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled

Note: In addition to the above response codes, the Control Sink can also send appropriate response codes described in IETF RFC 7231 [11] as applicable

7.6 Session termination

Example: DELETE /flus/v1.0/sessions/{session-res-id}

To terminate a FLUS session, the Control Source shall use the HTTP DELETE method on the "session" instance resource as follows:

- the request URL with the "apiPath" is set to "/flus/v1.0/sessions/{session-res-id}"
- the Host field is set to the FQDN of the FLUS sink

The {session-res-id} is the session resource identifier for which the session termination is sought.

Upon reception of a HTTP DELETE request, the Control Sink will check to see if the session with the given session resource identifier exists at the given sink. If such a session exists, the Control Sink shall delete the session instance associated with the given session resource identifier. Further, the Control Sink responds back to the Control Source with a 200 success message along with the session resource identifier indicating that the session was successfully deleted. Alternately, if the session cannot be deleted, the Control Sink shall send a 403 Forbidden message. If the session was not found, or if the sink was not found, the Control Sink returns back with a 404 message.

The possible response messages from the Control Sink, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.2.2.2.4-1.

Table 7.6-1: Response status code, message, and contents for session deletion

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink shall send the session resource identifier of the session that is deleted
204 No Content	The request has succeeded	None
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the Control Sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

8 FLUS Security

8.1 IMS-based FLUS

Security functionality for IMS is specified at both the control and media planes. From the FLUS system perspective, it is described in clause 4.2.1 that the FLUS media/user planes contain both the Media session establishment and the subsequent media data transmission functions. In other words, in an IMS-based FLUS system, both the IMS control and media plane security functionalities are combined in FLUS media and F-U as shown in Figure 4.2-2. From a logical standpoint, security functionality pertaining to session establishment procedures in the IMS-based FLUS system corresponds to IMS control plane security, which in turn consists of access and core network related security functions. The former, on access security, is defined in TS 33.203 [8], while the core network security functionality is defined in TS 33.210 [9]. Media data transmission in the IMS-based FLUS system include RTP transport of continuous media content, and its associated security functionality is defined in TS 33.328 [10].

8.2 Non-IMS-based FLUS

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

9 Media Codec Profiles

9.1 General

This clause specifies media codec profiles recommended for use with the Framework for Live Uplink Streaming. However, as a framework, FLUS is flexible enough to support use of other codecs and encapsulations that are not specified in any of the profiles.

9.2 IMS-based FLUS

If the FLUS source supports the default profile for live uplink streaming it shall support encoding requirements specified in clauses 5.3.2, 5.3.4, and 5.3.5 of [23] and the following encapsulation:

If the transmission of video is supported, then the RTP payload format of H.265/HEVC as specified in RFC 7798 [24] shall be supported

If the transmission of speech is supported, then the:

- RTP payload format of EVS as specified in TS 26.445 [25] shall be supported
- RTP payload format of AMR-WB as specified in RFC 4867 [26] should be supported
- RTP payload format of AMR as specified in RFC 4867 [26] may be supported

If the transmission of audio is supported, then RTP payload format of eAAC+ as specified in RFC 6416 [27] shall be supported

9.3 Non-IMS-based FLUS

If the FLUS source supports the default profile for live uplink streaming it shall support the encoding and encapsulation requirements specified in clause 5.3 of [23].

If the FLUS source supports the 360 Virtual Reality profile for live uplink streaming it shall support the encoding and encapsulation requirements specified in clause 5.6 of [23].

10 Remote Control

10.1 General

The FLUS Source may support a remote control interface through the F-RC. The Remote Control Target in the FLUS Source may subscribe to one or more remote control channels. If the remote control interface is supported, the Remote Control Target in the FLUS Source and the Remote Controller shall implement the MQTT protocol as specified in [21] for remote control.

The Remote Control Target shall connect at the FLUS session start-up to the MQTT Broker and may provide authorization credentials at subscription.

The Remote Control Target shall subscribe at the FLUS session start-up to one or more MQTT topics to receive remote control commands. The MQTT Broker may restrict subscription to MQTT topics.

The Remote Controller shall connect to the MQTT Broker and provide authorization credentials. The MQTT Broker shall restict publication to only authorized clients using authorization rules.

If remote control is supported, the Remote Controller and the Remote Control Target shall be provisioned with the MQTT broker address. In the IMS-based FLUS, the "a=remote_control_url:" session level attribute shall be used to signal the broker's address. The URI scheme shall be a "wss" or "mqtts".

The MQTT Topics shall be structured as defined in Clause 9.2.

The MQTT Messages may be formated according to Clause 9.3.

10.2 Usage of MQTT

The Topic shall be formatted according to the following ABNF syntax:

Topic=session_id "/" protocol ["/" remote_controller_id] ["/" media_source_id]

The session_id shall be the SIP session identifier for the IMS-based FLUS (as provided by the Session-ID SIP header [22]) or the F_C session resource id for the Generic FLUS. Note that topic names are case sensitive.

The protocol shall be a globally unique identifier of the application remote control protocol. The protocol defines the format of the payload message.

The remote_controller_id shall be a unique identifier of the remote controller. A Remote Control Target in the FLUS Source may receive message from different Remote Control Targets.

The media_source_id shall contain the unique identifier of the media source in that session. When the media_source_id is omitted, then the message should be applied to all media sources of that FLUS source. In case no media_source_id is provided in the PUBLISH or SUBSCRIBE messages, then all media sources of the FLUS Source are targetted by that message.

A Remote Control Target shall at least subscribe to the "session_id/protocol/#" topic, enabling the reception of message from any authorized Remote Controller and for any Media Source.

10.3 RC Framing Message Format

The following message framing for information exchange (e.g. the request and response) using F-RC may be used when delivering a message.

The remote control message shall be formatted as a json [RFC8259] formatted object.

- The json object contains three name / value pairs:
  - The first name / value pair represents the start line of an HTTP message. The name is "startLine" and the value is string according to RFC 8259.
  - The second name / value pair represents the optional headers. The name is "headers". The value is a json array of json object.
    - Each json object of the array contains the key and the value of a single HTTP header.
    - The Host HTTP header may be omitted, since the WebSocket Connection is already established.
  - The third name / value pair represents the optional body. The name is "body" and it is formatted according to the Content-Type header field. The content type of the body object is defined by the individual API.

Annex A (informative): Example deployment scenarios for FLUS sub-functions

A.0 General

This informative Annex illustrates several deployment examples for FLUS sub-functions. There may be additional deployments possible. The intention is to show different co-location options.

The 3GPP network is located between the Media Source and the Media Sink sub-function. Leveraging network services such as Quality of Service (QoS) is possible but not described in these deployment scenarios.

FLUS defines a Framework for uplink streaming and additional functions and interfaces may be added by implementation to realize a full uplink streaming service. All FLUS sub-functions expose local, undefined APIs, which can be used by other defined or undefined functions. Additional, service specific interfaces are possible and not depicted.

The FLUS Control (F-C) and FLUS Remote Control (F-RC) interface may be extended by implementations with additional configuration parameters, for example for capture device configurations or post processing and distribution function configurations.

The service specific implementation secures the sequencing of different FLUS functions and the correct instance provisioning using the local APIs.

A.1 Deployment with minimal FLUS sub-functions

A.1.1 Introduction

This clause describes deployments with minimal FLUS sub-functions from the standpoint that only F-U and F-C end-points are instantiated.

A.1.2 Deployment with a Media Source and a Control Source co-located in the same device

This clause describes a deployment scenario whereby the Control Source and the Media Source are implemented in the same device. The Media Source is provisioned via a local API, which includes at least the selected F-U instantiation, the Media Sink ingest information and optionally other information.

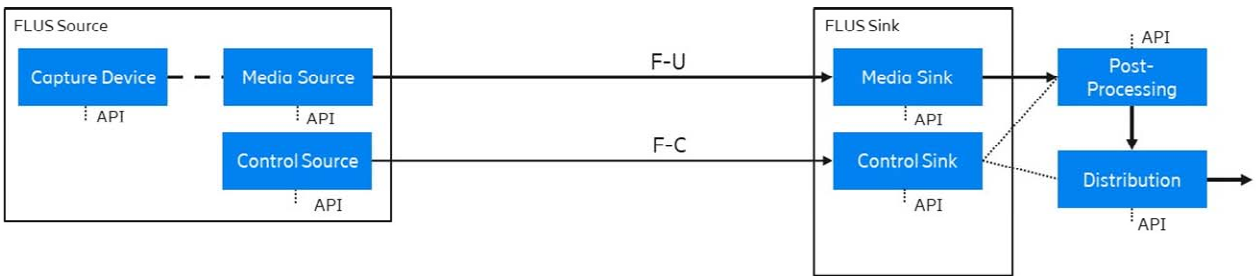


Figure A.1.2-1: Deployment with Minimal FLUS Subfunctions

It is assumed that the workflow is started from the Control Source. For example, via interaction with a User Interface (UI) and associated API, a human user causes the Control Source to set up and control the FLUS session. Additional application functions provision the Media Source (using the local API) according to the input to the Control Source.

A.1.3 Deployment with non-colocated Control Source and Media Source

This clause describes a deployment scenario whereby the Control Source and the Media Source are implemented in different functional entities. It is an example of the professional media production use case as described in TR 26.939 [22] whereby a production center facility, remotely located from the capture device, controls the establishment and termination of the FLUS session via F-C. The functional entity in which the Control Source is located is referred to as the CTRL entity.

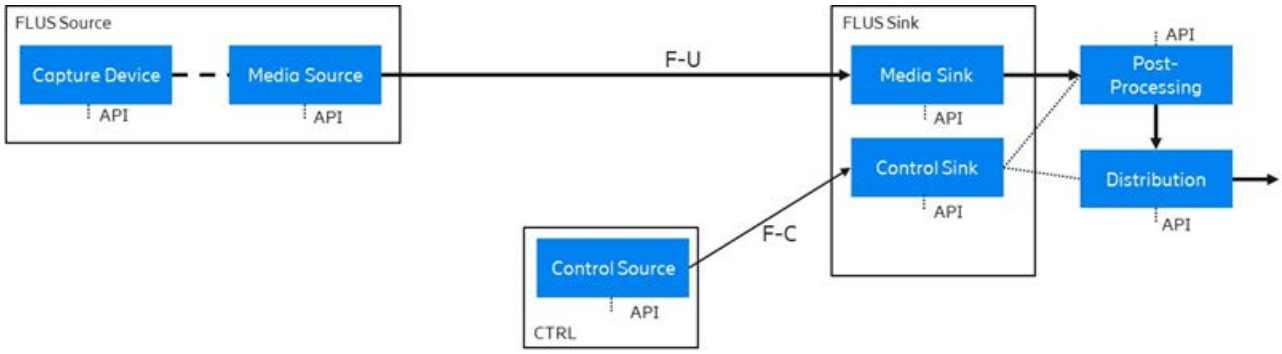


Figure A.1.3-1: Deployment with minimal FLUS sub-functions

It is assumed that the workflow is started from the Control Source. For example, via interaction with an UI and associated API, a human operator causes the Control Source function to establish and control the FLUS session. It is assumed that the Media Source has a local GUI so that it can be provisioned, via the selected F-U instantiation, the Media Sink ingest information and other implementation-specific configurations.

A.2 Deployments with FLUS remote control sub-functions

A.2.1 Introduction

This clause describes deployment scenarios whereby FLUS remote control sub-functions are present. Examples of implementations corresponding to the depicted system architectures are described in the sub-clauses below.

A.2.2 Deployment with stand-alone Control Source and Remote Controller sub-functions

This clause contains a deployment, where the Control Source and the Remote Controller are deployed on the same device (called CTRL device). The Remote Control Target is co-located with the Media Source on the same device (the FLUS Source). The architecture as shown in Figure A.2.2.1-1 is an example of the drone-mounted camera deployment scenario whereby a human user operated remote controller device performs the remote control of the drone camera for video capture and upstream delivery, as well as establishment/termination of the FLUS session, via F-C, in which media content transmission subsequently takes place via F-U.

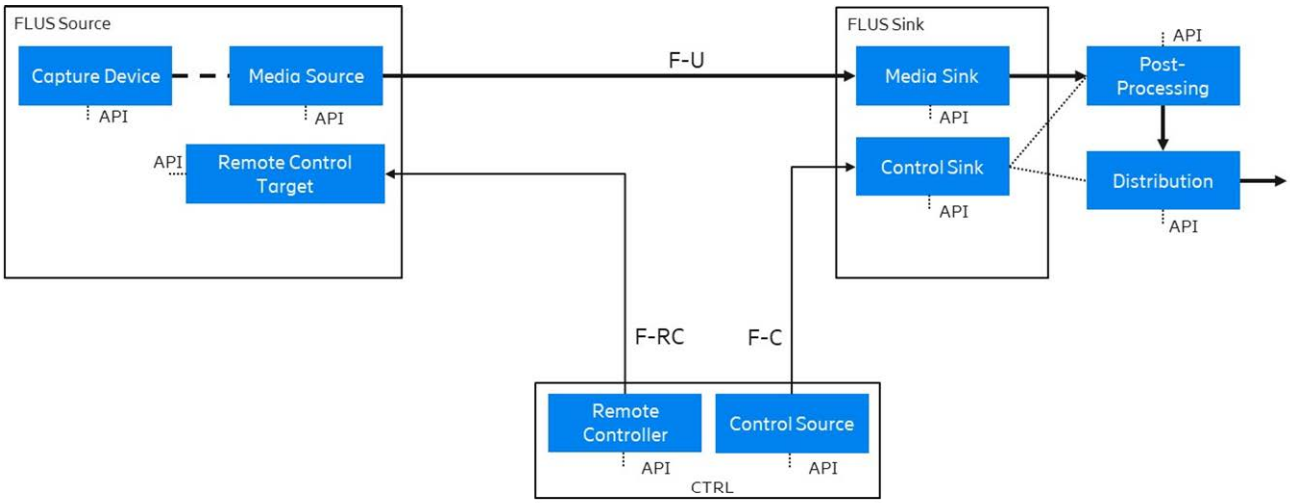


Figure A.2.2-1: Deployment with stand-alone Control Source and Remote Controller sub-functions

The CTRL device is able to communicate with both the FLUS Source and the FLUS sink. The CTRL entity is the primary function in performing configuration and control of the FLUS Source and FLUS Sink. The FLUS Source is configured at least with the address of the Remote Controller function within the CTRL device.

The Remote Control Target is provisioned with the address of the Remote Controller. Once the F-RC session is established, the FLUS Source can be provisioned via F-RC.

The F-U instantiation and the Media Sink selection is configured via F-RC. The F-RC interface can provide additional, implementation specific properties to support the provisioning of the Capture Device.

A.2.3 Deployment with a Remote Controller co-located with a Control Sink sub-function

This clause describes a deployment scenario whereby the Remote Controller is co-located with the Control Sink on the same functional entity. The Remote Control Target is deployed with the Media Source on the same device (the FLUS Source). It is an example of the professional media production use case as described in TR 26.939 [22] whereby a production center facility, remotely located from the capture device, controls the establishment and termination of the FLUS session via F-C, as well as performs remote control of the media capture equipment at a separate location (e.g., the event venue of a sports match or musical performance).

The FLUS Source is provisioned with the Remote Controller information via a local API. The FLUS Source receives the provisioning information of the selected F-U instantiation and the Media Sink ingest information via the Remote Control Interface.

It is possible to deploy FLUS Sink functions within a public cloud. The FLUS sink can be configured by a CTRL device, which is located behind a NAT / Firewall, such as within an enterprise network or within a PLMN. The FLUS Source functions are located on a wireless device, which is deployed within a PLMN. Traffic routing policies forbid the direct communication between the FLUS Sink and the CTRL entity.

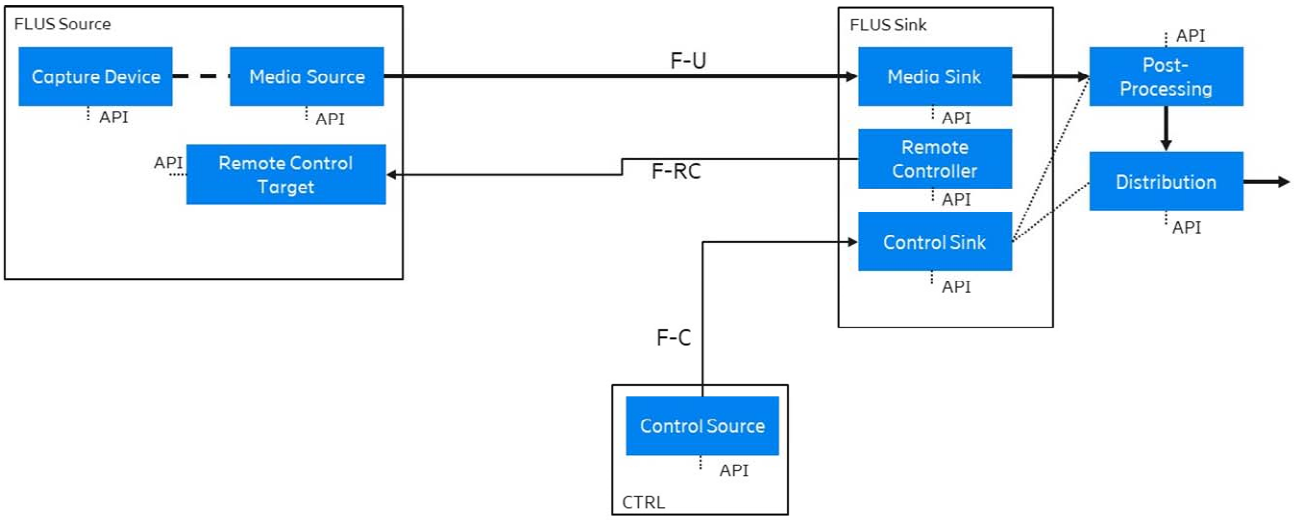


Figure A.2.3-1: Deployment with a Remote Controller co-located with a Control Sink sub-function

It is assumed, that the CTRL device is used as the main control function. FLUS defines a framework which supports the extension of existing F-C or F-RC interfaces with implementation configurations.

It is assumed the service specific implementation extends the F-C information with additional FLUS Source configuration information, so that the CTRL device can be the origin of the FLUS Sink configuration and the FLUS Source configuration. The FLUS Source configuration is relayed by an implementation-specific function from the Control Sink into the Remote Controller using local APIs.

Annex B (informative): Network-Based Media Processing (NBMP)

B.1. Introduction

MPEG NBMP in ISO/IEC 23090-8 [17] defines a framework that enables initializing and controlling media processing in the network. An NBMP Source describes the required media processing including the input and output formats and protocols, the required processing and configuration information. Based on this request, the NBMP Workflow Manager establishes the media processing workflow and informs the NBMP Source that the workflow is ready to start the processing. The Media Source(s) can then start transmitting their media to the network for processing.

NBMP control plane covers the following APIs (REST resources):

– Workflow API is used by NBMP Source to request NBMP Workflow Manager create and run a media processing workflow

– Function Discovery API provides the means for Workflow Manager and/or NBMP Source to discover media processing Functions that can be instantiated as part of a media processing Workflow.

– Task API is used by the NBMP Workflow Manager to configure, manage and monitor a Task at runtime.

The Workflow Description Document (WDD), Task Description Document (TDD) and Function Description Document (FDD) are exchanged using the above APIs as JSON objects.

On the media plane, NBMP describes the media formats, the metadata, and the supplementary information formats between the NBMP Source and the Task, as well as between the Tasks themselves.

Figure B.1 depicts the NBMP reference architecture depicting the above control plane and media plane interactions.

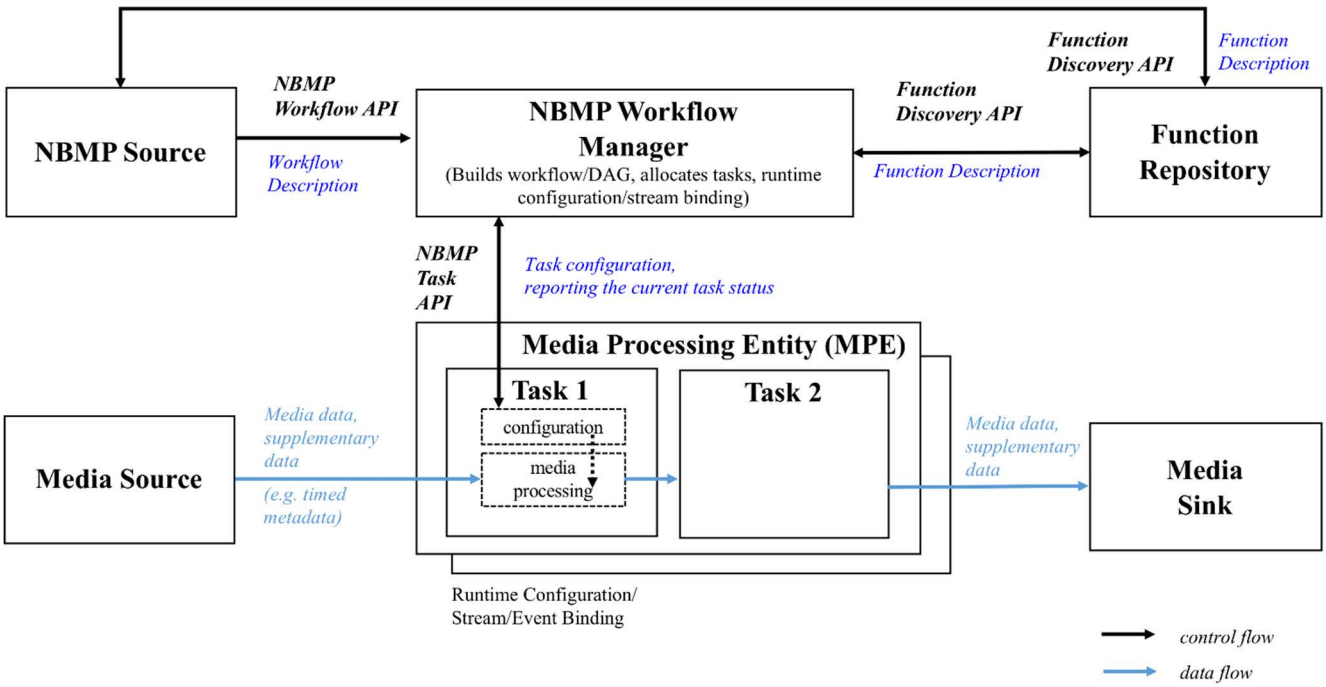


Figure X.1 – NBMP reference architecture

NBMP can be used for initiating media processing workflows such as 360 stitching, guided transcoding, overlaying on image or video backgrounds or view-dependent content customization.

B.2. NBMP Use in FLUS

The MPEG NBMP [17] Workflow Description Document (WDD) may be used to describe the media processing tasks at the FLUS sink to be performed on received media components from the FLUS source. As described in clause 5.3.6, NBMP WDD may be signalled as part of FLUS Sink Configuration properties.

Annex C (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-12	SA#78	SP-170884				Presented to TSG SA#78 (for approval)	1.0.0
2017-12	SA#78					Approved for Release 15 at TSG SA#78	15.0.0
2018-03	SA#79	SP-180024	0001	3	F	Corrections to FLUS Framework	15.1.0
2018-12	SA#82	SP-180974	0004	3	F	Architecture for QoS	16.0.0
2019-06	SA#84	SP-190341	0003	6	B	Remote Assist / Control Interface	16.1.0
2019-09	SA#85	SP-190651	0009	3	F	FLUS Remote Control Procedures	16.2.0
2019-12	SA#86	SP-190995	0011	2	F	Correction of FLUS Media Indication	16.3.0
2019-12	SA#86	SP-190995	0012	2	F	Update, correction, and clarification of reference and text	16.3.0
2019-12	SA#86	SP-190995	0013	1	C	Removal of AsInf Feature	16.3.0
2019-12	SA#86	SP-190995	0014	1	B	Uplink Assistance via RAN Signaling	16.3.0
2019-12	SA#86	SP-190995	0015	-	B	Amended QoS Signaling	16.3.0
2020-03	SA#87-e	SP-200037	0016	1	F	Removal of deprecated clauses	16.4.0
2020-03	SA#87-e	SP-200037	0017	1	C	NBMP Use in FLUS	16.4.0
2020-03	SA#87-e					Editorial Changes	16.4.1
2020-03	SA#87-e					Editorial Changes	16.4.2
2020-06	SA#88-e	SP-200389	0018	1	D	Correction of References	16.5.0
2020-06	SA#88-e	SP-200389	0021	-	B	Media Codec Profile for E_FLUS	16.5.0
2020-06	SA#88-e	SP-200389	0022	-	F	Clarification on uplink assistance	16.5.0
2020-06	SA#88-e	SP-200389	0023	-	F	Update, correction and clarification of text	16.5.0
2020-06	SA#88-e	SP-200389	0024	1	B	Adding Support for Remote Control	16.5.0
2020-09	SA#89-e	SP-200662	0025	2	F	Correction of FLUS F-C Stage 3	16.6.0
2020-10	Post SA#89-e					Update of Change History Table	16.6.1
2020-10	Post SA#89-e					Attaching back the file - 3GPP FLUS MO.xml	16.6.2
2021-12	SA#94-e	SP-211349	0026	1	C	CR on the extending support for network-based media processing in FLUS	17.0.0
2024-03	-	-	-	-	-	Update to Rel-18 version (MCC)	**18.0.0**

History

Document history		
V18.0.0	May 2024	Publication