|  |  |
| --- | --- |
| 3GPP TR 26.804 V0.1.0 (2021-02) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Study on 5G media streaming extensions  (Release 17) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2019, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

Introduction 5

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 6

3.3 Abbreviations 7

4 Examples for styles 7

4.1 Heading styles 7

4.2 Other common styles 7

Annex <X> (informative): Change history 9

# Foreword

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

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

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

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

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

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

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

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

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

**can** indicates that something is possible

**cannot** indicates that something is impossible

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

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

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

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

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

In addition:

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

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

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

# 1 Scope

The present document …

This Technical Report identifies and evaluates a set of potential improvements and extensions, referred to as key topics. The key topics are

- Content Preparation

- Traffic Identification

- Additional / New transport protocols

- Uplink media streaming

- Background traffic

- Content Aware Streaming

- Network Event usage

- Per-application-authorization

- Support for encrypted and high-value content

- Scalable distribution of unicast Live Services

For each of the above key topics, the following objectives are identified:

1. Document the above key topics in more detail, in particular how they relate to the 5GMS Architecture and protocols.

2. Study collaboration scenarios between the 5G System and Application Provider for each of the key topics.

3. Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

4. Map the key topics to basic functions and develop high-level call flows.

5. Identify the issues that need to be solved.

6. Provide candidate solutions (including call flows) for each of the identified issues.

7. Coordinate work with other 3GPP groups e.g. SA2, SA3, SA5, and others as needed.

8. Coordinate work with external organizations such as DASH-IF, CTA WAVE, ISO/IEC JTC29 WG3 (MPEG Systems), or IETF, as needed.

9. Identify gaps and recommend potential normative work for stage-2 call flows and possibly stage-3.

# 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] Akamai Blog, "A QUICk Introduction to HTTP/3", April 2020, <https://developer.akamai.com/blog/2020/04/14/quick-introduction-http3>

[3] Fielding, R., Nottingham, M., and J. Reschke, "HTTP/1.1", Work in Progress, Internet-Draft, draft-ietf-httpbis-messaging-13, 14 December 2020, http://www.ietf.org/internet-drafts/draft-ietf-httpbis-messaging-13.txt

[4] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, May 2015, https://www.rfc-editor.org/info/rfc7540

[5] draft-ietf-quic-http-33, "Hypertext Transfer Protocol Version 3 (HTTP/3)", 15 December 2020

[6] D. Bhat, A. Rizk, and M. Zink, "Not so QUIC: A Performance Study of DASH over QUIC," NOSSDAV'17: Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and VideoJune 2017 Pages 13–18 https://doi.org/10.1145/3083165.3083175

[7] AWS, "Achieving Great Video Quality Without Breaking the Bank", Streaming Media June 2019, [https://pages.awscloud.com/rs/112-TZM-766/images/GEN elemental-wp-achieving-great-video-quality-without-breaking-the-bank.pdf](https://pages.awscloud.com/rs/112-TZM-766/images/GEN%20elemental-wp-achieving-great-video-quality-without-breaking-the-bank.pdf)

[8] Netflix, "Optimized shot-based encodes: Now Streaming!", Netflix Blog, May 2018, https://netflixtechblog.com/optimized-shot-based-encodes-now-streaming-4b9464204830

[9] DASH-IF/DVB Report on Low-Latency Live Service with DASH, July 2017, available here: <https://dash-industry-forum.github.io/docs/Report%20on%20Low%20Latency%20DASH.pdf>

[10] DASH-IF IOP Guidelines v5, Low-latency Modes for DASH, available here: <https://dash-industry-forum.github.io/docs/CR-Low-Latency-Live-r8.pdf>

[11] ISO/IEC 23009-1, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats"

[12] IETF RFC 8673, "HTTP Random Access and Live Content".

# 3 Definitions of terms, symbols and abbreviations

This clause and its three subclauses are mandatory. The contents shall be shown as "void" if the TS/TR does not define any terms, symbols, or abbreviations.

## 3.1 Terms

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

Definition format (Normal)

**<defined term>:** <definition>.

**example:** text used to clarify abstract rules by applying them literally.

## 3.2 Symbols

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

Symbol format (EW)

<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].

Abbreviation format (EW)

<ABBREVIATION> <Expansion>

# 5 Key Topics

## 5.1 Introduction

## 5.2 Content Preparation

## 5.3 Traffic Identification

## 5.4 Additional/new transport protocols

### 5.4.1 Description

Media streaming applications are continued to use HTTP-based distribution protocols, but newer versions of HTTP such as HTTP/2 or HTTP/3 are introduced, see for example also TR 26.925 [5], clause 6.1.4. The architectural and performance impacts of such protocols for 5G-based media distribution is unclear and requires study. The study also considers how Media Players may use functionalities existing in new transport protocols, and also investigate the impact of new transport protocols on 5GMS usage and traffic identification (e.g. Service Data Flow Descriptions).

Based on [X], HTTP protocol (also known as web protocol), powers most websites, mobile apps, and videos. It was created by Tim Berners-Lee at CERN in 1989 and has been enhanced over the years to keep up with the ever-changing World Wide Web. Currently, the web is a mixture of HTTP/1.1 [3] and HTTP/2 [4] adoption. Most well-known websites are running HTTP/2, while smaller websites and late adopters plan to migrate to HTTP/2 in the near future as it is relatively easy to implement. HTTP/2 is used by about 45% of websites and supported by all major web browsers. HTTP/3 is only used by about 5% of websites now and not well-supported by web browsers yet. However. significant HTTP/3 deployments are emerging. For example, YouTube™ has for a long time been offering a pre-RFC draft version to any client that wants to use it, especially the Chrome™ browser. Other browsers are expected to follow soon after waiting for the QUIC and HTTP/3 RFCs to be published before mainlining that feature.

HTTP/2 provides on average a 5% to 15% performance improvement on page load times over HTTP/1.1. HTTP/1.1 allows persistent TCP connections, but requests still had to be serialized, resulting in the well-known "HTTP head of queue blocking". In order to improve downloads, many TCP flows still needed to be parallelized to speed up delivery.

HTTP/2 introduces the "Streams" concept at HTTP level and each stream can have different priorities. All objects can from a web-page can be multiplexed in single long-lived TCP connection. Also, HTTP/uses header compression (HPACK) to avoid verbose/clear text. Also, HTTP/2 pseudo-mandates TLS to prevent “middle boxes” from messing up with the content. However, HTTP/2 does not remove the drawbacks of TCP’s head-of-line blocking - packet loss on one stream will block all other streams until recovery even if packets for all other streams are correctly received.

HTTP/2 testing shows [2] that the delivery of large objects over HTTP/2 can be slower than over HTTP/1.1 when there is packet loss. This is because HTTP/2 uses a single TCP connection, versus about six connections which most web browsers open over HTTP/1.1. In addition, the TCP congestion control algorithms reduce the TCP congestion window size, resulting in fewer bytes sent over the wire when using just one TCP connection.

The solution to this problem is to use HTTP/2 over a different transport protocol that provides more efficient congestion control. One option would be to upgrade and modify TCP. Replacing TCP still needs to be checked carefully. For example, middle boxes such as NAT, Firewalls, Load balancers are problematic, they get rarely upgraded which prevents any updates to TCP. TCP is also hard to evolve as it is tied to OS Kernel. Hence, it was considered easier to introduce transport functions on top of UDP in the user space – referred to as QUIC.

That, in essence, is what HTTP/3 is: HTTP/2 over User Datagram Protocol (UDP) based on IETF QUIC. HTTP/3 is a thin layer on top of QUIC including QPACK header compression. The main QUIC functions are connection and stream multiplexing, fast startup, TLS1.3 (messages), loss recovery, in-order delivery (within stream), congestion control and flow control.

By multiplexing multiple concurrent logical streams over a single UDP-based transport association, and by giving each stream its own independent loss detection and recovery context, packet loss in one stream does not block progress on other logical streams in the same QUIC connection. (However, the affected stream will still block when packets are lost, so as to guarantee in-order delivery of payloads to the application.).

A screenshot of a cell phone

Description automatically generated

Figure 5.4-1: HTTP/2 and HTTP/3 protocol stacks

For an entertaining introduction to QUIC and HTTP/3, please check <https://www.youtube.com/watch?v=B1SQFjIXJtc>.

However, using QUIC for adaptive streaming still requires study as under certain circumstances, the quality using QUIC may even degrade for DASH-based streaming than it would increase [6]. The evaluation results show that using the unmodified DASH algorithms on top of QUIC may not provide the anticipated performance boost when compared to the standard DASH over TCP.

The main expected benefit of QUIC is being able to multiplex requests for all Adaptation Sets onto the same transport association, and then to manage the network QoS on that aggregate connection. This has a valuable operational benefit to a CDN operator (including the 5GMS AS) in reducing the number of UDP ports that a server needs to keep open. Another benefit is being able to migrate connections from one IP address to another with minimal interruption to either client or server. This is useful when the client moves, but it is also useful when the server changes (e.g. in edge computing relocation Use Cases).

### 5.4.2 Collaboration Scenarios

A service provider/content provider runs an adaptive media streaming service between HTTP/3 and QUIC enabled 5G Media Streaming AS and an HTTP/3 and QUIC enabled UE using 5G Media Streaming over M2d and M4d.

Editor’s Note: Study collaboration scenarios between the 5G System and Application Provider for each of the key topics.

### 5.4.3 Deployment Architectures

Editor’s Note: Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

### 5.4.4 Mapping to 5G Media Streaming and High-Level Call Flows

Editor’s Note: Map the key topics to basic functions and develop high-level call flows.

### 5.4.5 Potential open issues

Editor’s Note: Identify the issues that need to be solved.

### 5.4.6 Candidate Solutions

Editor’s Note: Provide candidate solutions (including call flows) for each of the identified issues.

## 5.5 Uplink media streaming

## 5.6 Background traffic

## 5.7 Content-Aware Streaming

### 5.7.1 Description

Content-Aware Encoding and statistical multiplexing of services are important and relevant technologies in the media industry. The impacts and opportunities of such technologies for 5GMS is not fully understood and requires study. For example, the currently-defined 5GMSd Application Function (AF) based network assistance solution is exclusively triggered by the Media Player, which instructs the Media Session Handler to interact with the Network. It might be more efficient for such network assistance functionality to be obtained directly from the content provider based on dynamic content complexity. Greater interaction with the 5GMS Application Provider during the lifetime of a session should be studied.

According to [2], if one analyses, almost any movie or television show scene by scene, you’ll notice the content has varying needs in terms of its fundamental complexity. Scenes with a lot of action and detail need a lot more bits in order to hit a quality target, whereas other scenes—say, a newsreader delivering a monologue—can achieve the same quality target with a reduced number of bits.

As an example, a game sequences provided for XR Traffic was encoded with x265 over 1 minute in Figure 5.7‑1. One can see that at the same quality, the number of bits required to represent the content can be quite different.

Figure 5.7-1: Bit rate and quality over time for an example sequence.  
(Blue bits, red PSNR in dB × 100)

Ideally, to maintain quality, one wants the bit rate to vary over time to maintain consistent quality regardless of the complexity of the scene. Four different scene types may be considered, and they differ in complexity- easy, moderate, hard, and very hard to compress. The “very hard” content might be a panning shot over a crowd, a shot of confetti falling, or simply a scene with a lot of high motion. Scenes such as these require more bits to convert all the motion and detail into a high-quality output that can be decoded and recreated accurately. A moderate scene, perhaps a close-up of a car, or an easy scene, like a single person speaking with no camera movement, will require fewer bits to deliver the same quality target as the harder scenes. In order to most efficiently encode the entire video, ideally a rate control mode that allocates more bits to the complex scenes, and fewer bits to the easier ones.

Different rate control algorithms exist:

* **CBR:** Constant-Bit Rate encoding keeps the bit rate at a constant level, but the quality fluctuates. In ancient systems such as MPEG-2 TS, this is even addressed by sending lots of filler data just to keep the pipe constant
* **VBR:** Variable Bit Rate encoding following the principle from above to keep the quality constant. This is often also referred to as Content-Aware Encoding nowadays (CAE).
* **Capped VBR:** in this case the basic idea is to ensure that you have a mix of the above, i.e. a certain bit rate is never exceeded, but in case the content does not need the data rate, less data is sent.

The below diagram attempts to address and show these issues, but is more confusing then helpful.



Figure 5.7-2: Capped VBR rate control encoding compared with CBR and VBR

Variance in bit rates for different users may also result on the device and the consumption model. For a smaller screen, quite likely quality and bit rate requirements can be lower than for example going to a large screen such as a 4K TV.

The 3GPP QoS model contradicts this, as typically resources and QoS parameters are assigned for a session and only GBR is addressed.

From a adaptive bitaret streaming perspective, this content model needs to also be viewed as part on the streaming model, as the complexity of the content may be addressed based on the buffer availability, and also the situation of the network needs to be studied.

1) On-demand Streaming:

a. Stationary streaming of C-VBR/CBR content: Typically one operates with receiver buffer levels of 5-30 seconds [check details in TS 26.512]. One tries to keep the buffer filled. As soon as your buffer drains below some threshold, the client triggers a down-switch to a more sustainable bit rate. Switching typically can happen at segment boundaries, for example every 2 seconds.

b. Start-up and seek. In this case, one starts basically starts from an empty buffer. In order to have quick and stable start up and good quality right away, there may be a benefit to get a higher short-term bit rate from the network to fill the buffer quicker to at least the switching threshold as you would not start playback until the threshold is reached. This is a very instantaneous action and needs to be fulfilled instantanteously, at most after 1 second.

c. Stationary streaming of VBR/CAE content: In this case you basically operate on buffers of 5-30 seconds as above. The client typically has a map of the bit rate over time profile. In this case the client knows how much bit rate it needs for the next 5-10 seconds in order to keep the buffer stable and it can provide this information in a continuous manner to the network. The network will then grant a certain bit rate. This aspect may be fulfilled with using existing 5GMS functionalities.

2) Live Streaming and especially low-latency live streaming:

a. General: In this case the buffer is something of duration 1-5 seconds, it can be kept really low for low-latency streaming. Typically, one operates e2e latency of 3-5 seconds, so the buffer in the client is low. In addition, the client does not know the exact bit rate of the content as it is produced on the fly. Switching can typically be done every 1-2 seconds

b. Stationary streaming CBR: the buffer is much more susceptible, and you may have a threshold of maybe 500 ms when the client needs a fast arriving Segment is not arriving fast enough. This aspect may be fulfilled with using existing 5GMS functionalities.

c. Start-up is similar to on-demand streaming as your buffer is anyways low. So no difference

d. Yet another and probably the most interesting case is the live case, for which the content and each of Representations are VBR encoded, but more following the content complexity and VBR/CAE is done as shown below. The content complexity is not known in advance, but it needs to be provided on an content ingest interface to the network. In this case, the network should provision very fast and dynamically the bit rate if needed, but can relax.

There are many other cases, where content complexity and device characteristics need to be taken into account when addressing quality of service.

### 5.7.2 Collaboration Scenarios

In the following, difference collaboration scenarios are provided. In Figure 5.7-3, content is generated by a third-party content provider in different formats and configurations, taking into account for example:

1. Different device types (resolution, frame rates, codecs).
2. Different streams (target qualities and bit rates).
3. Encoding parameters (CBR, VBR, etc.).



Figure 5.7-3: Content-Aware Streaming based on static parameters

The network and client can make use of this information in order to optimize the streaming.

In a second variant, not only static information is provided, but also dynamic information with the media stream. This data is provided from the content provider to the 5G Media Streaming system and the client.



Figure 5.7-4: Content-Aware Streaming based on dynamic parameters

The client can use the information for:

* Optimizing its own quality.
* Acting fairly in a way that it only requests higher bit rates when the content is more complex, but leaves remaining capacity to the community.

In an On-Demand service, the use of the information is client controlled, i.e. the network is unaware of content complexity.

* Client downloads a description of the content variations and the associated quality initially.
* Client now brokers with the network for an average bit rate, but ability to request higher bit rate when content is complex
* DASH client includes logic to use the VBR options smartly to ask for “boosts” ahead of time when content is complex

Live Streaming and Ingest (network more actively included in streaming):

* Encoder provides as early as possible indication that content for same quality is getting a complex.
* Network uses this and identifies, if and how to fulfill this for the clients that request it.
* To not confuse client throughput estimation, communication between network and client is necessary.

### 5.7.3 Deployment Architectures

Editor’s Note: Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

### 5.7.4 Mapping to 5G Media Streaming and High-Level Call Flows

Editor’s Note: Map the key topics to basic functions and develop high-level call flows.

### 5.7.5 Potential open issues

Editor’s Note: Identify the issues that need to be solved.

### 5.7.6 Candidate Solutions

Editor’s Note: Provide candidate solutions (including call flows) for each of the identified issues.

## 5.8 Network Event usage

## 5.9 Per-application-authorization

### 5.9.1 Description

[Operation of certain 5GMSA and 5G System enabled services include an SLA between the Application Provider and the 5GMS System provider. Different solutions to enable per-application authorization should be studied.]

Editor’s Note: Document the above key topics in more detail, in particular how they relate to the 5GMS Architecture and protocols.

### 5.9.2 Collaboration Scenarios

Editor’s Note: Study collaboration scenarios between the 5G System and Application Provider for each of the key topics.

### 5.9.3 Deployment Architectures

Editor’s Note: Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

### 5.9.4 Mapping to 5G Media Streaming and High-Level Call Flows

Editor’s Note: Map the key topics to basic functions and develop high-level call flows.

### 5.9.5 Potential open issues

Editor’s Note: Identify the issues that need to be solved.

### 5.9.6 Candidate Solutions

Editor’s Note: Provide candidate solutions (including call flows) for each of the identified issues.

## 5.10 Support for encrypted and high-value content

### 5.10.1 Description

Content is increasingly encrypted for distribution for different reasons, e.g. Content Protection, Conditional Access, or integrity of playback. The management of keys for different use cases is a prime concern. Examples include scalable access to keys, secure storage of keys, key availabilities. It is envisioned that an MNO can provide key management and/or key distribution services for content providers. In particular, providing scalable and secure key management within 5GMS for multiple different devices needs further study.

Examples for secure media specification are for example provided by the MovieLabs ECP requirements and other content providers requirements.

In a specific example, a live sports service provider wants to offer a live stream. Examples include where the content needs to be delivered with low latency (typically encoder to glass in 3 – 10 seconds) in order to be on par with regular TV distribution means. Other services may also be considered.

The service may require different tools and functionalities levels of security:

1. Conditional access supported by DRM management. As an example, users need to get a master key for decrypting the secondary level keys.
2. Key rotation in order to support live streaming. As an example, these keys are changed periodically but protected by the master key.
3. DRM and key management to ensure playback rules, for example to avoid that clients attempting early playback of the content too early and have advantages in betting/wagering, skipping content, etc.
4. Watermarking: The content is distributed and a unique signature is added at the latest possible time (in the device, at the Edge). An example of such approach can be found here <https://learn.akamai.com/en-us/webhelp/adaptive-media-delivery/adaptive-media-delivery-implementation-guide/GUID-3F89E64C-415D-452D-9541-BB650CD783B9.html>.
5. Content encryption.
6. A secure implementation (use of TEE, Secure Media Path).

### 5.10.2 Collaboration Scenarios

It is assumed that the content provider provides DRM protections for the content. However, beyond this different collaboration models between the content provider and 5G System operator/MNO exist.

As examples, the MNO provides infrastructure to the content service provider in order to support security related functions.

- The service provider may want to provide scalable access to the content and in particular the key distribution. Hence it uses 5G Media streaming servers to support secure key distribution.

- The streaming service provider wants to rule playback, for example to avoid that the situation whereby users can see the streamed content too early while at the same time, the streaming service provider does not want to delay the distribution artificially either and want to give the clients the ability to download the main content (without buffer underruns).

- The service provider asks for fairness in the client, but the client cannot be trusted to act fairly. Hacked clients are possible. Clients may have DRM systems that the service providers will use.

- The service provider asks for a watermarking solution from the MNO.

Encryption (as already defined in TS 26.511 [3]) and secure keys may be used for other purposes, for example for conditional access or DRM systems. In some cases, keys are also provided in hierarchically, depending on business rules, security levels and deployment scenarios.

In an extension of the above use case, the content is distributed via multiple operators network. In this case, the encryption may be done by the service provider and the service provider provides the keys to the MNO. In another case, the service is offered by the MNO and the MNO does encryption and key management.

Editor’s Note: Study collaboration scenarios between the 5G System and Application Provider for each of the key topics.

### 5.10.3 Deployment Architectures

Editor’s Note: Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

### 5.10.4 Mapping to 5G Media Streaming and High-Level Call Flows

Editor’s Note: Map the key topics to basic functions and develop high-level call flows.

### 5.10.5 Potential open issues

Editor’s Note: Identify the issues that need to be solved.

### 5.10.6 Candidate Solutions

Editor’s Note: Provide candidate solutions (including call flows) for each of the identified issues.

## 5.11 TV-grade mass distribution of unicast Live Services

### 5.11.1 Description

Live TV services of different scale (professional, user-generated, session-based, etc.) are increasingly distributed over broadband and mobile networks. Live TV services are characterized by:

- scalability (in terms of concurrent users),

- consistent quality,

- high bandwidth requirements, and

- target latency constraints.

Consistent support of the distribution of such services to a different scale of users and in a concurrent fashion is a prime concern. 5G Media Streaming is expected to support such service distribution and end-to-end optimizations. Improvements and optimizations on the architectural level and stage 3 are expected to be studied.

Based on a report developed jointly between DVB and DASH-IF on Low-Latency DASH [9], this clause defines details on how to support consistent latency in DASH for linear TV services. In [9], several definitions had been introduced, repeated here for consistency.

*- End-to-End Latency (EEL)*: The latency for an action that is captured by the camera until its visibility on the remote screen.

*- Encoder-Display Latency (EDL)*: The latency of the linear playout output (which typically serves as input to distribution encoder(s)) to the screen.

*- Packager-Display Latency*: The latency after the output of the distribution encoder to the screen.

*- CDN latency*: The delay caused by the CDN delivery from CDN input to CDN output.

*- Live Edge Start-up Delay (LSD)*: The time between a user action (service access or service join) and the time until the first media sample of the service is perceived by the user when joining at the live edge. Typically also the channel change time.

*- Seek Start-up Delay (SSD)*: The time between a user action (service access or service join) and the time until the first media sample of the service is perceived by the user when seeking to a time shift buffer.

Those two categories, latency and delay are subject to be controllable by the service provider for a consistent service offering. In the remainder, primarily the Encoder-Display Latency (EDL) and the Live Edge Start-up Delay are considered, but for some use cases also the End-to-End Latency (EEL) may be relevant. Figure 5.11.1‑1 provides a schematic overview of the different latencies.



Figure 5.11.1‑1: Different latencies and delays relevant for low-latency distribution

The Low Latency DASH scenario is a variant of the Live Services recommended approach focused on ensuring that the Encoder-Display Latency of the DASH Media Presentation is comparable to the latency when distributing over terrestrial, cable or satellite broadcast. Latency in broadcast is not a unique universal value, as it is influenced by many factors such as the duration of the broadcast encoding pipeline, the latency of the transport channel which can slightly differ per type (satellite, cable, IPTV or, DTT...), or the artificial delays introduced by local content moderation regulations. However, most of the measurements converge on a 3 - 10 seconds latency between the moment where the source signal is acquired for encoding and the moment when it's played back on the TVs, i.e the EDL. Start-up delay requirements are typically in the range of 1-2 seconds. For details refer to [9].

Low-latency mode are supported to minimize the architectural impacts on existing workflows. Figure 5.11.1‑2 provides a basic flow of information for operating a low-latency DASH service as defined in DASH-IF’s Low-latency Modes for DASH [10]. The DASH packager gets information on the general description of the service as well as the encoder configuration. The encoder produces CMAF chunks and fragments. The chunks are mapped by the MPD packager onto Segments and provided to the network in incremental fashion using HTTP/1.1 chunked transfer.



Figure 5.11.1-2 Basic operation flow Low-Latency DASH

HTTP chunked transfer coding needs to be supported up from the ingest into the packager up to the CDN edge, whereas the last mile delivery is expected happen using HTTP chunked transfer coding or HTTP in regular mode. If HTTP chunked transfer coding is supported by the DASH player, it basically means that a media segment carrying the latest moment of the program (also known as the "live edge time" as defined in clause 4 of this document) could be consumed on the player while it's still being produced by the encoder and the packager.

In case chunked segments are used, clients may want to access partially available Segments for example for fast random access, see ISO/IEC 23009-1 [11]. However, requesting available byte ranges of a partially available Segment, i.e., Segments still being produced, is not consistently supported in CDNs, but solutions are provided in RFC8673 [X6]. This functionality may also be needed to support common segment handling for low-latency DASH and low-latency HLS.

Key aspects for low-latency live distribution include:

*-* Consistent support for chunked transfer from ingest to client.

*-* Support for partially access of non-complete resources.

*-* End-to-end optimizations to support the latency requirements.

### 5.11.2 Collaboration Scenarios

Editor’s Note: Study collaboration scenarios between the 5G System and Application Provider for each of the key topics.

### 5.11.3 Deployment Architectures

Editor’s Note: Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

### 5.11.4 Mapping to 5G Media Streaming and High-Level Call Flows

Editor’s Note: Map the key topics to basic functions and develop high-level call flows.

### 5.11.5 Potential open issues

Editor’s Note: Identify the issues that need to be solved.

### 5.11.6 Candidate Solutions

Editor’s Note: Provide candidate solutions (including call flows) for each of the identified issues.

Annex <X> (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| Jan 2021 | SA4#112 |  |  |  |  | Initial version | 0.0.1 |
| Feb 2021 | SA4#112 | S4-210054  S4-210056 S4-210298 S4-210302  S4-210303 |  |  |  | Key Topic Content Aware Streaming  Key Topic Per-application-authorization  Key Topic Additional / New transport protocols  Key Topic Support for encrypted and high-value content  Key Topic Scalable distribution of unicast Live Services | 0.1.0 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |