## **3GPP TSG-SA WG4 Meeting #130 *S4-242145***

**Orlando, Florida, USA 18-22 November 2024** revision of ***S4-242026***

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  | **26.804** | **CR** | **0018** | **rev** | **1** | **Current version:** | **18.1.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

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

|  |
| --- |
|  |
| ***Title:***  | KI#15 Dynamic content generation from multiple sources |
|  |  |
| ***Source to WG:*** | Huawei Hisilicon |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | FS\_AMD |  | ***Date:*** | 2024-11-12 |
|  |  |  |  |  |
| ***Category:*** | **B** |  | ***Release:*** | Rel-19 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | Introduce the key issue of dynamic content generation in redundant workflows that is common in the media industry. There are some benefits in deploying this closer to the user in the mobile network. |
|  |  |
| ***Summary of change:*** | Dynamic content generation from distributed sources is common in the media industry, by adopting dynamic content generation in the network more targeted and personalized experiences can be achieved. Examples include forensic watermarking, advertisement insertion, additional codec or DRM support and many other use cases could be enabled. The baseline case of resilience and redundancy is also significantly beneficial.As of today, 5G media streaming is not (widely) adopted in the industry.This additional feature may make it more attractive to enable media streaming related functionality in the network using 5GMS.  |
|  |  |
| ***Consequences if not approved:*** | FS\_AMD will not enable new and improved user experiences, potential of 5G media streaming is not utilized. |
|  |  |
| ***Clauses affected:*** | 2, 5.x (NEW) |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | * Address qualcomm comments on API usage
* Address BBC comments on structure
* BBC restructure
* Replace diagram and add KI description
* Add conclusions and recommendations
 |

## FIRST CHANGE

## 2 References

…

[DASH9] Draft Text of ISO/IEC FDIS 23009-9 Information technology - Dynamic adaptive streaming over HTTP (DASH) – Part 9: Redundant Encoding and Packaging for segmented live media (REaP), ISO/IEC JTC 1/SC 29/WG 3 NO 1165, Jan. 26, 2024. [Online]. Available: <https://www.mpeg.org/standards/MPEG-DASH/9/>

[26510] 3GPP TS 26.510: "Interactions and APIs for provisioning and media session handling (Release 18)".

[MHV02] Roberto Ramos-Chavez, Espen Braastad, Jamie Fletcher, and Arjen Wagenaar. 2024. Standards based OTT A/B Watermarking at Scale. In Proceedings of the 3rd Mile-High Video Conference (MHV '24). Association for Computing Machinery, New York, NY, USA, 94. <https://doi.org/10.1145/3638036.3640251>

## SECOND CHANGE (ALL NEW TEXT)

## 5.x KI#15 Dynamic content generation from multiple sources

### 5.x.1 Description

#### 5.x.1.1 Introduction

Media service providers may obtain content from multiple sources. Examples include:

1. Advertising content is inserted into live content that is distributed
2. Content targeted at a specific user is inserted into content in certain periods of time.

3. Content is obtained from multiple sources to have redundancy in the operation.

In order to integrate the content into a conforming Media Presentation – typically CMAF-based with an MPEG-DASH and/or HLS manifest – the content needs to fulfil certain conditions. One way to address the issue is the use of transcoding at the combination point of the content from multiple sources. However, such a process is costly and time-consuming. Hence, it is preferable that the content that needs to be combined into a conforming media presentation is produced in a way such that it can be combined to produce conforming content in a dynamic fashion without transcoding.

1. Content from different sources needs to be interchangeable (not necessarily identical). This is not a problem for static content, but for dynamically generated content such as live media this can be problematic because differences may occur during various system configuration such as clock synchronization issues or delay in different parts of the workflow, but also general configuration settings related to the media processing to dynamically generate the presentation.

2 When content is generated closer to the user (e.g., at the edge of the network) more dynamic and interesting media presentations can be realized as a more localized version of the media can be generated. Dynamic content generation in the network can be attractive for operators.

This study considers integration of different technologies into the 5G Media Streaming System that addresses the case of dynamic content generation at different points in the content distribution chain, and similar issues by allowing media streaming applications to efficiently generate and access content that may be changing dynamically at the edge of the network from different content serving endpoints. Different client implementations may then beneficially use the content from these multiple sources or networks concurrently, potentially guided by the service or by the network provider.

This both improves redundancy (resilience to failures) and flexibility (extending media services).

In addition to be being able to dynamically generate content from multiple endpoints for redundancy/resilience, it can also enable use cases such as content replacement, introducing new codecs and/or DRM systems (as each endpoint can generate representations of a different codec or DRM system). Another application is watermarking where multiple sources generate otherwise identical content, but the embedded watermark is different, and both segments and playlists need to be interchangeable.

The key aspect addressed is the use of redundant workflows with multiple encoding and/or packaging steps. In this case, a single contribution source passes through multiple distribution encoders and packagers that may be deployed throughout the network.

#### 5.x.1.2 Challenges in Multi-endpoint dynamic content generation

Dynamic content generation usually includes different components such as contribution encoders, distribution encoders, DRM encryptors etc. This is to be able to create an optimized and targeted experience for the end user. By generating content closer to the end user in the network, more targeted and personalized experience can be achieved.

Challenges in deployments with multi-endpoint deployments and architectures with dynamic content generation may include:

1. *Sustained CDN-/network-wide service disruptions* where network access, connectivity or QoS is severely degraded. Examples may include cases where an entire CDN’s network is degraded because of a network-wide misconfiguration or power failure. In this case a client should switch to another CDN for retrieving the content.

2. *Intermittent or short-term disruptions or delay between upstream components.* Examples include short periods of congestion within the network, isolated HTTP request/response failures or delays caused by servers that may lead to different state at different downstream servers.

3. *Timeline issues with the content.* Due to slight time differences, the playlists or manifests may list a different number of segments or have a different live edge. When a player switches from one endpoint to another, it may therefore result in retroactive timeline changes and may potentially be unable to continue playback. These changes may happen because of different delays on different network paths, or slightly different configurations on a particular path.

4. *Segment replacement and substitution at the edge requires aligned timelines and segment durations to be successful.* Guidelines for appropriate content formatting and generalized content formatting may be needed. This can enable redundant segment generation for resilience, watermarking, content replacement etc.

#### 5.x.1.3 Dynamic Content Publishing from multiple endpoints

When streaming media presentations originate from different sources, consistency of the media presentations is important.

This is trivial to achieve for the case of static non-changing content.

However, live media streaming presentations are updated frequently, resulting in updated presentation manifests (e.g., MPEG-DASH MPD), updated media playlists (HTTP Live Streaming), and different segments becoming available or no longer being available.

The support for such live streaming cases is important for multi-endpoint scenarios, because it is desirable that both media segments and presentation manifests are interchangeable even when they originate from different endpoints.

In practice, CDN source A may have a slightly delayed version of a presentation compared to CDN source B, resulting in different segment availability, and potentially conflicting presentation manifests and/or media playlists.

Such timing differentiations may happen upstream in the workflow due to different delays or differently configured upstream components. Even if the delays are relatively modest, discrepancies between sources may still occur that are disruptive to the user experience.

Practice has shown that players of streaming media presentations are sensitive to such issues. For example, retroactive changes to the media timeline may cause playback failure.

Figure 5.x.1.3-1 shows the example case of a retroactive timeline changes caused by switching from one endpoint to another that has a delayed version of the dynamic media presentation.



Figure 5.x.1.3-1: Sample case of playback failure in multi-CDN delivery

The corresponding example sequence of events is as follows:

1. The media presentation description is received with given MPD@publishTime, earliest media presentation time EPT, last segment presentation time LPT in the DVR window.

2. Playback starts.

3. After an MPD update request the DVR window is updated without problems because the update is consistent since it comes from the same sources.

4. In a subsequent request, the media player switches to an alternative endpoint AS 2 (or CDN source B) and this has delayed input.

5. An updated MPD is received with a later publish time, but the earliest presentation time of the newest segment is decreased and the timeline is changed retroactively.

6. As a consequence, playback is interrupted because the media player could not handle this.

Other related issues that may occur due to different source versions may be different timeline gaps in the media presentation and overall differences in the segment availability between subsequent updates.

#### 5.x.1.4 Examples of inconsistent Content Publishing from multiple sources (CDN)

This clause provides some example scenarios in which inconsistent timelines can occur in dynamic content publishing, even if the segments are interchangeable and both source contents have an identical timeline. The cause can be that content publishing origin B receives a delayed input compared to content publishing origin A.

Each dynamic content publishing origin generates the media playlist description based on its input and sets the publishTime based on the system clock time that is synchronized to global timing systems.

In this case the first MPD is retrieved from origin A that is ahead, in this case the newest segment time is:

1031634228096 + 1152 × (312 + 1) = 1031634588672

Then the player receives the MPD from content publishing origin B in a delayed data centre it receives an MPD with a later publish time, but the start time of the newest segment is:

1031634226944 + 1152 × (312 + 1) = 1031634587520

Which is earlier, this implies that the newest segment from the MPD from origin A is not the MPD, but the publishtime is later, leading to a retroactive timeline change interpreted by the media player.

Listing 5.x.1.4-1 Snapshot of publishTime and segmentTimeline
in MPD retrieved from origin A

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?><MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd" type="dynamic" availabilityStartTime="1970-01-01T00:00:00Z" publishTime="2024-06-26T08:36:24.332711Z" minimumUpdatePeriod="PT2S" timeShiftBufferDepth="PT10M" maxSegmentDuration="PT2S" minBufferTime="PT10S" profiles="urn:mpeg:dash:profile:isoff-live:2011,urn:com:dashif:dash264"> <Period id="1" start="PT0S"> <AdaptationSet …… <SegmentTemplate timescale="600" initialization="live-$RepresentationID$.dash" media="live-$RepresentationID$-$Time$.dash"> <!-- 2024-06-26T08:26:20.160000Z / 1719390380 - 2024-06-26T08:36:21.120000Z --> <SegmentTimeline> <S t="1031634228096" d="1152" r="312" /> </SegmentTimeline>……</MPD> |

Listing 5.x.1.4-2: Snapshot of publishTime and segmentTimeline
in MPD retrieved from origin B

|  |
| --- |
| <MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd" type="dynamic" availabilityStartTime="1970-01-01T00:00:00Z" publishTime="2024-06-26T08:36:25.378162Z" minimumUpdatePeriod="PT2S" timeShiftBufferDepth="PT10M" maxSegmentDuration="PT2S" minBufferTime="PT10S" profiles="urn:mpeg:dash:profile:isoff-live:2011,urn:com:dashif:dash264"> <Period id="1" start="PT0S"> <AdaptationSet …. <SegmentTemplate timescale="600" initialization="live-$RepresentationID$.dash" media="live-$RepresentationID$-$Time$.dash"> <!-- 2024-06-26T08:26:18.240000Z / 1719390378 - 2024-06-26T08:36:19.200000Z --> <SegmentTimeline> <S t="1031634226944" d="1152" r="312" /> </SegmentTimeline>….</MPD> |

#### 5.x.1.5 Redundant Encoding and Packaging (REaP)

Redundant Encoding and Packaging (REaP), as defined in MPEG-DASH Part 9 [DASH9], specifies a Delivery Media presentation Description (D-MPD) which adds constraints on the formatting and output generation of the media presentation description to make sure that outputs form different distributed origins application servers are interchangeable by a player (i.e. no retroactive timeline changes). These constraints apply to the representations, the setting of MPD@publishTime and some other attributes/elements in the MPD. A corresponding version for HTTP Live Streaming playlists is defined as well.

NOTE: REaP is not an API but rather an architecture to support existing segmented media streaming formats and additional constraints on formats to enable use cases with dynamic content generation.



Figure 5.x.1.5-1: Architecture for redundant encoding and packaging

The REaP architecture depicted in figure 5.x.1.5-1 starts from defining a reference/example head-end architecture, that in the case of 5G Media streaming may be the 5GMS AS or even components upstream of the 5GMS AS. The key assumption is that a common contribution source (encoder) with a common timeline is used that can be converted back to a timeline relative to Unix epoch, that is to say midnight UTC on 1st January 1970. This assumption can hold in many cases where a common contribution signal is used in the streaming head-end. This way all tracks and streams have the same timeline origin. Secondly a regular media segmentation strategy is proposed resulting in aligned media segment boundaries of the different segments.

This is used in the output of one or more distribution encoders that produce ISO BMFF tracks with aligned segment boundaries and a common shared timeline relative to Unix epoch. In addition, REaP defines the I-MPD, a constrained version of the media presentation description to announce the streams that is for example used for live ingest or distribution encoder egress.

The critical part are the distributed packagers in combination with the origin as this corresponds to the output generated at the content origin or multiple content origins in the multi-source dynamic content generation case.

For this case REaP defines the Delivery MPD (D-MPD) that constrains the formatting of the output MPD by linking some of the fields in DASH MPD that can cause retroactive changes to the media timeline. For example, by linking the MPD@publishTime explicitly to the earliest presentation time of the newest segment, and constraining the way the set of available segments are updated, it is possible to generate consistent media presentation description or HTTP live streaming playlists.

The approach in REaP can work in modestly well synchronized content origins up to 100 ms out of sync. A key aspect for a REaP setup is to configure the expected maximum delay incurred in the workflow/AS to account for this setting MPD@publishTime as a function of the earliest media presentation time of the latest segment at the live edge.

The approach has, for example, been demonstrated using an open source implementation [MHV02] to implement a pseudo-watermarking workflow. In pseudo-watermarking, the two content origins create interchangeable content that only differs by an embedded watermark. In this demo the watermark was shown visually on the screen, showing how the player is able to seamlessly play content generated dynamically from different sources.

In REaP this problem case shown in figure 5.x.1.5.1-1 is overcome by coupling the publish time to the media timeline in the output to enable consistent dynamic content publishing between sources. If the publish time was coupled to the media time this specific case of retro-active timeline change could have been avoided, but it requires some further configuration.

Therefore, some configurations from REaP may be considered in 5GMS, such as segment duration, the epoch used, the synchronization time stamp for linking the publish time and media timeline.

#### 5.x.1.6 Key Issue objectives

This Key Issue aims to tackle the following questions:

1. How can dynamic content generation from multiple sources be integrated into the 5GMS System in order to address the problem of dynamic content generation from multiple sources.

a. Where in the 5GMS architecture does the generation of content segments occur?

b. Where in the 5GMS architecture does the generation of presentation manifests occur?

c. Is the Content Preparation feature of the 5GMSd AS fit for purpose in this context?

2. Review use cases enabled by dynamic generation from multiple sources.

a. Explore and document use cases for generating content dynamically in the trusted network.

b. Improving redundancy and failover in 5GMS deployments using dynamic content generation from multiple sources and configuring this in 5GMS.

c. Develop call flows and workflows for such cases of dynamic content generation from multiple sources.

### 5.x.2 Collaboration scenarios

#### 5.x.2.1 Multi-endpoint media delivery with dynamic content generation

In this scenario, the 5GMSd Client requests adaptive media streaming content from two or more 5GMSd Application Server endpoints. The 5GMSd Client may choose one 5GMSd AS endpoint or use multiple simultaneously. This allows the client to distribute network load across Application Server instamces and M4 downlink transports, optimize costs, as well as improve QoS.

The content retrieved over reference point M4d includes both media segments and manifests/playlists. In some cases, a different 5GMSd AS endpoint may serve different groups of segments and/or different media playlists/manifests.

The 5GMSd Client’s Media Session Handler discovers the URLs of these 5GMSd AS endpoints from the 5GMSd Application Function (5GMSd AF), either through a Media Entry Point or from a separate piece of metadata. QoE metrics from the 5GMSd Client may be used by the 5GMSd AF to determine the best 5GMSd AS instance(s) for each client to use when streaming media. Figure 5.x.2.1-1 shows the 5GMSd Client communicating with multiple 5GMSd AS instances. Each 5GMSd AS instance has no direct communication with its peers; rather it communicates (minimally) with the 5GMSd Application Provider via reference point M2d and with the 5GMSd AF (not depicted) via reference point M3d.



Figure 5.x.2.1-1: Multi-AS media delivery within 5G system with dynamic content generation

In this case the 5GMSd AS includes dynamic content generation and generates, based on content ingested from the 5GMSd Application Provider via reference point M2d, new media segments and/or representations.

### 5.x.3 Architecture mapping

#### 5.x.3.1 Content formatting at multiple 5GMSd AS endpoints following REaP

REaP as defined in [DASH9] may be used in the context of the downlink media streaming architecture by instantiating a streaming head-end in the 5GMSd AS. To create an interchangeable media presentation independently at each 5GMSd AS instance, the content preparation feature is used as shown in the below figure.



Figure 5.x.3.1-1: Possible deployment scenario with multiple 5GMsd AS implementations

### 5.x.4 High-level call flow

Editor's Note: A high-level call flow will be provided once this CR is endorsed. Generally REaP is useful for configuring content preparation, the high level call flow is not expected to be different from general high level call flow for content preparation.

### 5.x.5 Gap analysis and requirements

Editor's Note: Possible need to standardise a Content Preparation Template format to configure the REaP streaming head-end via the 5GMSd AS content preparation feature. Could an existing media manipulation configuration format be adopted and profiled by 5GMS, e.g. from MPEG NBMP?

Editor's Note: What do the Media Entry Points passed to the 5GMS Client look like for REaP? Are they passed at M1+M5 or M8 or either?

### 5.x.6 Candidate solutions

#### 5.x.6.1 Content formatting at multiple 5GMSd AS endpoints following REaP

This solution proposes to solve the issue of dynamic content generation from multiple sources by using additional format constraints that can be applied to DASH and HLS streaming media presentations (both media segments and media playlist or media presentation description). These constraints are based on the Redundant Encoding and Packaging ISO/IEC 23009-9 from MPEG (ISO/IEC SC 29 WG 3) and no new formats envisioned to be defined in 3GPP. Also REaP formats from ISO/IEC 23009-9 are a subset of current formatting options available) so it is more an enhancement to enable use cases with multiple encoders/packagers for live content production.

This solution is based on Redundant Encoding and Packaging (REaP) as defined in MPEG-DASH Part 9 [DASH9]. The solution defines formats for usage in the streaming head-end that typically resides in the 5GMSd AS enabling generation of interchangeable media presentations at different sources. The formats in REaP are based on existing MPEG formats such as MPEG-DASH, ISO BMFF and Common Media application with some additional constraints to enable interchangeable formats. Aspects include consistent segment durations, shared timeline (originating from epoch).

Proposed standardization work if for further study and could include the following:

At stage 2, the recommendation is to update TS 26.501 [15] to include the use case of redundant encoding and packaging in the description of the content preparation use case.

At stage 3, the recommendation is to add some REaP-specific features to TS 26.512 [16], namely:

- REaP formatting enablement at the 5GMSd AS using its content preparation feature.

- Configuration of REaP parameters for the content preparation (e.g. Segment Duration, REaP synchronization timestamp P-STS).

The advantage of integrating REaP into the 5GMS architecture is that changes are limited, and the output format from the 5GMSd AS can be easily checked for conformance using existing conformance tooling. In addition, very limited or up to no signalling is needed between 5GMSd AS instances, making the solution very easy to adopt and maintain in practice.

### 5.x.7 Summary and conclusions

Support for ReAP and configuration of REaP in 5GMS is for further study, some possible directions are given in the candidate solution in 5.x.6.1.

## END OF CHANGES