**3GPP TSG-SA WG4#114e S4-210xxx**

**Electronic Meeting, 19th-28th May, 2021**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **Pseudo CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.804** | **CR** |  | **rev** | **-** | **Current version:** | **0.2.1** |  |
|  | | | | | | | | |
| *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 |  | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Update on the content aware streaming | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Huawei Technologies Co.,Ltd. | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_5GMS\_EXT | | | | |  | ***Date:*** | | | 2021-5-10 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **B** |  | | | | | ***Release:*** | | | <Release> |
|  | *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 can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Another type of content aware streaming is completed. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Add another type of content aware streaming. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The relevant architecture components are not complete and misleading. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.3.3 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  |  | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  |  | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  |  | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

## 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 5.7.1‑1. One can see that at the same quality, the number of bits required to represent the content can be quite different.

Figure 5.7.1-1: 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 an adaptive bit rate 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.

For example, there is another kind of content-aware streaming where different media content may have different importance from the user experience perspective. The media content within one media streaming service therefore place different requirements on the media transport network.

1.- For viewport-dependent streaming, the streaming contains two parts: one for the low resolution omnidirectional segments and the other is for the high-quality Field-Of-View, as shown in figure 5.7.‑3 below. When the two parts of media content are requested, the low-resolution omnidirectional segment may be of higher importance in order to guarantee the basic user experience. It takes less time to display the required media content.

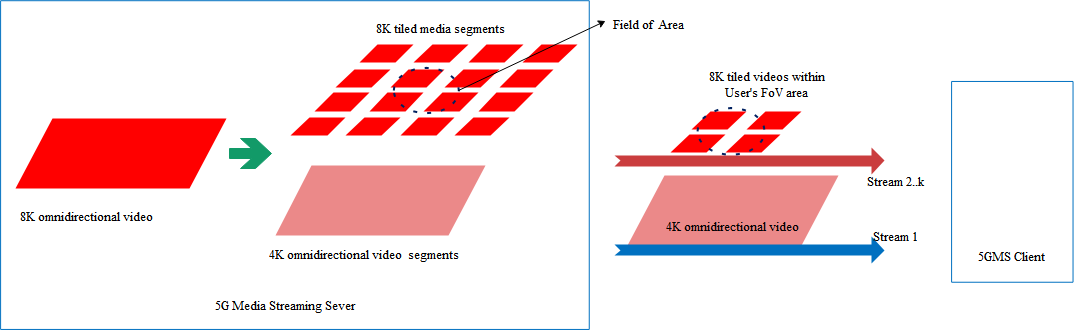


Figure 5.7‑3: media streamings for viewport dependent streaming scenario

2. A similar use case can also be found in the FS\_XRTraffic work item [X]. In real-time XR services, e.g. cloud gaming and split rendering, the frames/slices may have difference importance from the user experience perspective. For example, an encoded I-frame has higher importance because the subsequent frames cannot be correctly decoded without it, as illustrated by the dependency graph in figure 5.7‑4 below. Conversely, the last frame which is not referenced by other frames is the least important.

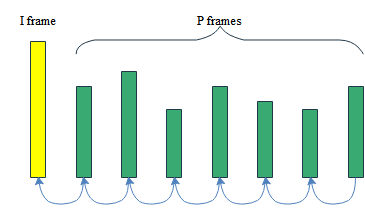


Figure 5.7‑4: Inter-frame references within one GOP

3. Finally, content importance may also vary according to the type of source media content. For example, when there is more than one content type within one a streaming session, e.g. video content, audio content and subtitles, the audio content may be much more important than the video and subtitles in maintaining a minimum media experience for the end user.

Due to the different network requirements, different media content within a media streaming session may need to be handled in a different way, for example by applying different QoS configurations.

### 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.

According to the different importance of different media content, the 5GMS Client or 5GMS AF can indirectly or directly invoke the dynamic policy APIs to apply different QoS configurations to different media contents. The high-level call flows based on the 5G Media Streaming architecture can be shown as in figure 5.7.4‑1 below.



NOTE: Potential enhancements to step 3, 4, 5 and 6 for QoS related mechanisms may need further work in SA2.

Figure 5.7.4‑1: High-level call flow for differential QoS treatment of 5G Media Streaming 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.