## **3GPP SA4#129-e *S4-241588***

**Online, 19th - 23rd August 2024** *revision of S4al240114*

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  | **26.804** | **CR** | **0006** | **rev** | **9** | **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:***  | [FS\_AMD] Multi-CDN and Multi-Access Media Delivery |
|  |  |
| ***Source to WG:*** | Dolby France SAS |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | FS\_AMD |  | ***Date:*** | 2024-05-23 |
|  |  |  |  |  |
| ***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:*** | Document the FS\_AMD key topic “c) Multi-Access and Multi-CDN Delivery” description and collaboration scenarios. |
|  |  |
| ***Summary of change:*** | Addition of new clause 5.x Multi-CDN and Multi-Access Media Delivery including sub-clause structure and headings. Addition of prose for Description and Collaboration Scenarios. |
|  |  |
| ***Consequences if not approved:*** | FS\_AMD objectives not achieved. |
|  |  |
| ***Clauses affected:*** | 2, 5.x (NEW) |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  |  |
| ***affected:*** |  | **X** |  Test specifications |  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications |  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | S4-240591: SA4#127-bis-e, description and collaboration scenariosS4-240844: updates with comments received at SA4#127-e-bisS4aI240052: updated to match proposed template and address pending comments.S4al240894: S4al240052 endorsed at SA4 post 127-bis-e and comments/changes accepted for ease of reading. Updates with comments received at SA4 post 127-bis-e and introduction of candidate solutions. More background on candidate solution is contained in S4-240895. Multi-access media delivery overview and collaboration scenarios have been moved to S4-241082.S4-241230: updates to address pending comments.S4-241273: updates to address pending comments.S4al240095: S4-241273 endorsed at SA4#128. Comments/changes accepted for ease of reading. CMMF candidate solution architecture and reference point descriptions added.S4al240107: CMMF candidate solution architecture and reference point descriptions replaced with options to incorporate CMMF within the existing 5GMS architecture.S4al240114: Edits from BBC. Endorsed by MBS SWG. |

## 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/>

[UNPKG24] Emma Roth, "A popular open-source content delivery network went down for hours", The Verge, Apr. 12, 2024. [Online]. Available: <https://www.theverge.com/2024/4/12/24128276/open-source-unpkg-cdn-down> (accessed May 9, 2024).

[NET23] Sebastian Moss, "Cloudflare recovers from service outage after power failure at core North American data center", Data Center Dynamics, Nov. 3, 2023. [Online]. Available: <https://www.datacenterdynamics.com/en/news/cloudflare-recovers-from-service-outage-after-power-failure-at-core-north-american-data-center/> (accessed May 9, 2024).

[FSLY21] Brian Barrett, "How an Obscure Company Took Down Big Chunks of the Internet", Wired, Jun. 8, 2021. [Online]. Available: <https://www.wired.com/story/fastly-cdn-internet-outages-2021/> (accessed May 9, 2024).

[AKAM21] Josh Fomon, "CDN Provider Akamai Takes Down Popular Internet Services During Outage", Ookla, Jul. 22, 2021. [Online]. Available: <https://www.ookla.com/articles/akamai-outage-july-22-2021> (accessed May 9, 2024).

[NET22] Charlotte Trueman, "Cloudflare outage brings hundreds of sites, services temporarily offline", Computer World, Jun. 21, 2022. [Online]. Available: <https://www.computerworld.com/article/1627967/cloudflare-outage-brings-hundreds-of-sites-services-temporarily-offline.html> (accessed May 9, 2024).

[VZ19] Jim Salter, "The Internet broke today: Facebook, Verizon, and more see major outages", Ars Technica, Jul. 3, 2019. [Online]. Available: <https://arstechnica.com/information-technology/2019/07/facebook-cloudflare-microsoft-and-twitter-suffer-outages/> (accessed May 9, 2024).

[DEMX01] Marc Hoppner, "A content owner, a CDN and a player walk into a bar", (Jan. 6, 2023). Accessed: May 9, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=S9EdoQFOQ9I&list>
=PLkyaYNWEKcOf98lZxnCcL6y7ZIVU3oSYO&index=12

[DEMX02] Guillaume du Pantavice, "Improving streaming experience with Bayesian optimization, from AB to AZ test", (Dec. 25, 2021). Accessed: May 9, 2024. [Online Video]. Available: https://www.youtube.com/
watch?v=t4nRrLygVwo&list=PLkyaYNWEKcOfD1GYFxFbZXDP03XM-cZPg&index=19

[IEEE01] E. Ghabashneh and S. Rao, "Exploring the interplay between CDN caching and video streaming performance", IEEE INFOCOM 2020 – IEEE Conference on Computer Communications, Toronto, ON, Canada, 2020, pp. 516-525.

[ACM01] K. Vermeulen, L. Salamatian, S. H. Kim, M. Calder, and E. Katz-Bassett, "The central problem with distributed content: common CDN deployments centralize traffic in a risky way", In Proceedings of the 22nd ACM Workshop on Hot Topics in Networks (HotNets ’23). Association for Computing Machinery, New York, NY, USA, 70-78.

[MHV01] A. Bentaleb, R. Farahani, F. Tashtarian, C. Timmerer, H. Hellwagner, and R. Zimmermann, "Which CDN to Download From? A Client and Server Strategies", (Jan. 6, 2024). Accessed: May 9, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=xCZmCnWgQRE>

[VAS01] Will Law, "Content steering with MPEG DASH", (May 4, 2023). Accessed: May 9, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=o9Pa5y-Usxw>

[MWS23] W. Law and Y. Reznik, "DASH content steering at scale", Media Web Symposium (MWS’23), June 2023.

[DIFCS] ETSI TS 103 998: "DASH-IF: Content steering for DASH".

[CMMF] ETSI TS 103 973: "Coded multisource media format (CMMF) for content distribution and delivery".

[TS26501] 3GPP TS 26.501: "5G Media Streaming (5GMS) General description and architecture (Release 18)".

[TS26510] 3GPP TS 26.510: "Interactions and APIs for provisioning and media session handling (Release 18)".

[TS26512] 3GPP TS 26.512: "5G Media streaming (5GMS) protocols (Release 18)".

[RAPTORQ] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, “RaptorQ forward error correction scheme for object delivery,” IETF RFC 6330, August 2011.

[RSFEC] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo, “Reed-Solomon forward error correction (FEC) schemes,” IETF RFC 5110, April 2009.

## SECOND CHANGE

## 5.19 Multi-CDN media delivery

### 5.19.1 Description

#### 5.19.1.1 Introduction

Media streaming applications conventionally obtain content from a single source over a single path within a network. This imposes several limitations:

1. Performance is constrained to that of the source and path chosen. Whatever the limits on network bandwidth and latency between the client and that source are directly translated to the client’s achievable Quality of Service (QoS) and Quality of Experience (QoE).

2 Disruptions or degraded performance caused by the source in use or on any of the network links between the client and source can lead to poor user experience, often in the form of lower playback quality, rebuffering, or complete playback failure.

This study considers integration of different technologies into the 5G Media Streaming System that addresses these, and similar, issues by allowing media streaming applications to efficiently access content across multiple Content Delivery Networks (CDNs) and/or multiple access networks. Different client implementations may then beneficially use the content on these multiple sources or networks concurrently, potentially guided by service or network provider. In addition, formats and techniques for generating content for multiple CDN or multiple access network delivery such as MPEG-DASH Part 9 (ReAP) [DASH9] may be considered. Further extensions include the ability for a client to use multiple access networks at the same time to support media delivery. Study of integration of different technologies into the 5G Media Streaming System is of relevance to address content provisioning, content hosting, impacts on user plane reference points M2 and M4, and on media session handling at reference point M5 as well as potential benefits in terms of quality and resource usage.

#### 5.19.1.2 Challenges Multi-CDN deployments aim to address

CDNs are often used by content distributors to globally scale delivery of their content to end-users. These networks consist of a number of Points of Presence (PoPs) located at various locations around the networks’ edge. These PoPs help load-balance delivery of content as well as improve Quality of Service (QoS) by reducing the distance/latency between every client and the content they are accessing. In many cases, content distributors employ multiple CDNs to leverage the strengths of one CDN over another in every location those CDNs have a PoP. For example, a client experiencing degraded performance while using one CDN may switch to another that is offering better performance at that time and location. As another example, a content distributor may prefer one CDN over another at a given time to reduce delivery costs and/or meet monthly contractual commitments. These multi-CDN deployments aim to solve content delivery issues that exist when only one CDN is used; but the benefits they provide may not be fully realized because of the various challenges experienced and underlying methods used to stream content to every client.

Challenges multi-CDN deployments and architectures aim to address 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. The duration of these events may last minutes to hours and affect a majority of the client population. Examples of recorded instances can be found in [UNPKG24], [NET23], [FSLY21], [AKAM21], [NET22], and [VZ19].

2. *Intermittent or short-term disruptions affecting QoS for an individual or small group of clients.* Examples include short periods of congestion within the network, isolated HTTP request/response failures or delays caused by application server congestion, etc. The following discusses these in greater detail: [DEMX01], [DEMX02], [IEEE01], [ACM01], [MHV01], [VAS01], and [MWS23]

3. *Augmentation of one CDN's performance with that of another to achieve a level of performance that neither can provide on its own.* An example is a peer-to-peer CDN where each peer has limited uplink capacity and is unable to satisfactorily service client demand on its own.

#### 5.19.1.3 Coded Multi-source Media Format (CMMF)

Coded Multi-source Media Format (CMMF) is an extensible container format designed to facilitate the management and interchange of audio-visual media and metadata in one or more coded representations (e.g., encoded with application-layer, linear, network, or channel codes). The coded media representations supported by CMMF enable the efficient use of multi-source, multi-path, and multi-access connectivity for network-delivered media applications. The use of CMMF does not replace the basic media streaming architectures and procedures already defined. Rather, it is intended to supplement them to provide additional capabilities.

To understand some of the advantages of using CMMF for streaming media, CMMF was implemented and trialed on a commercial streaming platform from approximately September 2022 through September 2023. This platform offers a large content library, streamed to a world-wide customer base where the majority of the content had a maximum bit rate of 5 Mbps or less (the median maximum bit rate available was approximately 3.5 Mbps while over 70% of all sessions had a maximum possible bit rate of 5 Mbps or less). Approximately 5% - 50% of the traffic on selected device types was streamed using CMMF while the remainder of the traffic was streamed using a popular conventional server-side switching/DNS-based multi-CDN implementation. Both the CMMF multi-source and the conventional multi-CDN approach used three tier 1 CDNs. CMMF clients downloaded content from each CDN in parallel, while the "conventional" clients switched between the three based on input from the multi-CDN switching platform. Performance measurements for all traffic were collected using an industry-leading performance measurement platform. This data includes session-level information about relevant QoE key performance indicators (KPIs). In addition, supplemental QoS information was collected by the CMMF SDK for only those sessions using multi-source as a delivery method.

A summary of the amount of traffic measured for each delivery method during this trial is provided in Table 5.19.6.3-1. This and subsequent tables only show traffic measured for Android clients streaming over cellular networks from January 1 through July 26, 2023. Furthermore, only those sessions where the mean edge cache hit rate is greater than 50% are considered. For CMMF traffic, this was determined using the supplemental QoS information collected by the CMMF SDK for each session. For convenional traffic, no information was available on a session-by-session basis since this traffic bypassed the SDK. Rather, it was confirmed via querying each utilized CDN that the mean edge cache hit rates for all conventional traffic was greater than 95%. This estimate of the edge cache hit rate was also validated in a separate experiment where conventional traffic was routed through the CMMF SDK so that QoS metrics (including cache hit status) could be collected. Unfortunately, the volume of CMMF traffic and the diversity of the content streamed during the trial made it very difficult to keep CDN caches warm with CMMF encoded content. Trying to match multi-source and conventional edge cache hit rates on a one-to-one basis was not possible. As a result, the threshold established above provides sufficient data to provide statistically significant results; but it also implicitly favors conventional delivery since those sessions were more often served by the CDNs’ edge.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Delivery method | Hours watched | Number of playback sessions | Number of unique devices | Number of unique countries | Minutes watched per unique device |
| Conventional | 25,026.92 | 120,269 | 23,752 | 178 | 63.22 |
| CMMF | 14,013.76 | 44,081 | 12,534 | 141 | 67.08 |

Table 5.19.6.3-1: CMMF real-world multi-CDN trial summary. Only sessions measured on cellular networks and running Android are shown.

An overview of the performance improvements multi-source delivery provided over conventional multi-CDN switching for various QoE KPI’s is shown in Table 5.19.6.3-2. The table provides the mean value of the relevant KPI plus/minus one standard deviation. In general, double-digit gains were observed across all key QoE performance indicators showing that CMMF enabled multi-source delivery can drastically improve the quality of streamed media.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Delivery method | Normalized average session bit rate(% of max session bit rate)(NOTE 1) | Playback start-up time(s)(NOTE 2) | Connection-induced rebuffering ratio(%)(NOTE 3) | Start-up failure rate(%) | Playback failure rate(%) |
| Conventional | 83.70 ± 28.08 | 3.40 ± 10.08 | 0.28 ± 1.78 | 0.51 | 1.22 |
| CMMF | 94.31 ± 16.23 | 1.83 ± 9.34 | 0.19 ± 1.17 | 0.07 | 0.59 |
| Difference | 10.61 ± 32.43 | -1.57 ± 13.74 | -0.09 ± 2.13 | -0.44 | -0.63 |
| Gain | + 12.68% | - 46.18% | - 32.14% | -86.27 % | - 51.64% |
| NOTE 1: The normalized average session bit rate is defined as the average bit rate measured during a session divided by the maximum bit rate listed in the session’s content manifest. Only sessions without a start-up or video playback failure, a playing time greater than or equal to 60 seconds, playback completed at least 10% of the content, and the maximum bit rate as defined by the sessions’ corresponding manifest was available. These statistics are weighted using the sessions’ duration.NOTE 2: Only sessions without a start-up or playback failure and a start-up time greater than 0 seconds.NOTE 3: Only sessions without a start-up or playback failure, a playing time greater than or equal to 60 seconds, and playback completed at least 10% of the content. These statistics are weighted using the sessions’ duration. |

Table 5.19.6.3-2: Real-world multi-CDN QoE performance results.

The empirical CDFs for the content normalized average session bit rate, start-up time, and connection-induced rebuffing ratio are provided in figures 5.19.6.3-1, 5.19.6.3-2, and 5.19.6.3-3 respectively.

Figure 5.19.6.3-1 shows that 60% of the sessions, regardless of delivery method, experienced an average session playback bit rate close to the maximum possible based on the content being played. However, CMMF multi-source delivery was able to lift more of those clients that could not reach the highest bit rate further up the bit rate ladder than conventional delivery.



Figure 5.19.6.3-1: Empirical CDF of the content normalized average session bit rate.

Figure 5.19.6.3-2 shows that CMMF multi-source delivery was able to significantly reduce the video start-up time as well. For example, only 10.4% of the CMMF sessions experienced a startup time greater than 3 seconds compared to 29.0% of the conventional sessions.



Figure 5.19.6.3-2: Empirical CDF of the video startup time.

Finally, Figure 5.19.6.3-3 shows that CMMF multi-source delivery reduced the number of sessions that experienced a connection-induced rebuffering event from 22.4% to 14.9% in addition to reducing the total duration of rebuffering given a rebuffering event occurred.



Figure 5.19.6.3-3: Empirical CDF of the connection-induced rebuffering ratio (CIRR).

Multi-CDN and/or multi-access media delivery using CMMF requires preparation of the content to be served to a population of clients and clients that can access and download from multiple sources in parallel. Specifically,

1. The ability to create CMMF-encoded media objects and distribute/stripe these (in addition to possibly distributing the original source media such as MPEG-DASH or HLS media segments) across multiple client-accessible network locations (e.g., 5GMS Application Servers, CDNs, etc.).

 Accessing content from multiple sources within the network simultaneously requires that each network source be populated with a unique CMMF bitstream/object containing the content being requested. A CMMF network source is one that can be individually addressable or reachable (i.e., there should be a one-to-one mapping between the set of individually addressable or reachable sources and the set of CMMF bitstreams/objects for each CMMF encoded piece of content). Source types may be entire CDN distributions, single points-of-presence (PoPs) within a single CDN distribution, or standalone servers. For example, a single CDN which replicates content across their PoPs and uses DNS or anycast to route traffic to the PoPs within their network would be considered one source. Alternatively, a CDN that enables clients to reach individual PoPs within their network may allow for each PoP to be an CMMF source assuming each PoP can be populated with a unique CMMF bitstream/object.

 Various methods for creating unique CMMF bitstreams/objects for each CMMF network source exist. The necessary CMMF bitstreams/objects can be created offline (e.g., at the time the video/audio is encoded and packaged) and stored on an origin server for later retrieval by the CMMF network sources. They can also be created on demand using a cloud-based or edge-based just-in-time encoder as client requests are received.

 Detailed examples for preparing original source media for delivery from multiple serving endpoints using CMMF are provided in [CMMF]. In general, the processing required to create CMMF bitstreams/objects is minimal (and scalable) allowing for a wide range of available implementation options.

2. The capability for clients to access, efficiently download, and jointly decode partial CMMF-encoded media bitstreams/objects in parallel from multiple network locations where CMMF-encoded media objects (and possibly original source media) are stored/cached.

 These capabilities can be implemented as a plug-in or software development kit (SDK) to simplify integration into existing platforms and players, or they can be implemented directly within the streaming media player located on each client. When downloading content (e.g., a segment that is intended to be played), a CMMF client will connect to multiple sources and request the CMMF bitstream/object associated with that content from each. Any one of these CMMF bitstreams/objects do not need to be obtained in their entirety, nor does any byte-level scheduling need to occur (e.g., each CMMF bitstream can be transmitted from their beginning to their end). Rather, a client only needs to obtain enough information from all of the transmitted CMMF bitstreams/objects so that it can decode the content those bitstreams/objects carry.

 The methods used to efficiently download media using CMMF from multiple sources are heavily dependent on the underlying network and transport protocols used to deliver CMMF-encoded bitstreams/objects, as well as the implementation of the CMMF-enabled client. For streaming use cases using either HTTP/1.1 or HTTP/2 over TCP, total overhead (i.e., total amount of data egressed from all of the sources (including HTTP and CMMF headers) with respect to the size of the original source media object) has been demonstrated to be between 1-3% (it should be highlighted that this is roughly on-par with player ABR induced overhead related to downloading multiple bit rates of the same segment and (obviously) rendering only one during playback). Overhead when using other network/transport protocols or different implementations may differ.

 The complexity and client device impacts of jointly decoding received CMMF bitstreams/objects has also been demonstrated to be minimal. While the decode complexity is dependent on the CMMF code type used ([CMMF] supports a variety of different code types including general deterministic and random linear codes (RLC), the 3GPP Raptor code [RFC5053] as defined in TS 26.346 [26346], RaptorQ [RAPTORQ], and Reed-Solomon [RSFEC]), CMMF has been demonstrated on over 4,000 unique client device models without issue.

### 5.19.2 Collaboration scenarios

#### 5.19.2.1 Multi-CDN media delivery

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

The client’s Media Session Handler discovers the URLs of these Application Servers from the 5GMSd Application Function (AF), either through a Media Entry Point or from a separate piece of metadata. QoE metrics from the client may be used by the AF to determine the best Application Server(s) for each client to use when streaming media.

Figure 5.19.2.1-1 shows the client communicating with multiple Application Servers. Each AS has no direct communication with its peers; rather it communicates (minimally) with the Application Provider and with the 5GMSd AF (not depicted) via reference point M3d.



Figure 5.19.2.1-1: Multi-CDN media delivery within 5G system

#### 5.19.2.2 Joint multi-CDN and multi-access media delivery

In this scenario, the 5GMSd Client is directly connected to multiple data, or access, networks (e.g., an unmanaged Wi‑Fi network and the 5G network), as described in clause 5.18. The client requests adaptive media streaming content from two or more 5GMSd Application Servers. The Client may choose one or use multiple simultaneously. This allows the client to distribute network load across access networks and Application Servers, optimize costs, as well as improve QoS.

The client’s Media Session Handler discovers the URLs of these Application Servers from the 5GMSd Application Function (AF), either through a Media Entry Point or from a separate piece of metadata. QoE metrics from the client may be used by the AF to determine the best Application Server(s) for each client to use when streaming media.

Figure 5.19.2.2-1 shows the client communicating with multiple Application Servers through different data networks. Neither data network nor AS has direct communication with its peers. Rather each 5GMSd AS communicates (minimally) with the 5GMSd Application Provider at reference point M2 and with the 5GMSd AF (not depicted) via reference point M3d.



Figure 5.19.2.2-1: Multi-access media delivery within 5G system

### 5.19.3 Architecture mapping

#### 5.19.3.1 Server-side CDN switching

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.3.2 Client-side CDN switching

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.3.3 Concurrent CDN access using CMMF

Several options on the network side, client side and configuration exist when implementing CMMF within the 5GMS System. These are expanded upon further below.

##### 5.19.3.3.1 Network-side implementation of CMMF within the 5GMS architecture

The use of CMMF for delivering content from multiple sources/CDNs requires that the content be encoded into multiple coded representations (one per source/CDN). This requires the source content to be prepared (i.e., encoded within CMMF) somewhere within the network prior to a client attempting to access it. Various options exist for where these encoding procedures take place. These options include:

1. *5GMSd Application Provider.* In this option, it is the responsibility of the 5GMS Application Provider to encode and package source content within CMMF bitstreams/objects prior to delivery of that content separately to each 5GMSd AS instance via reference point M2d or to each external 5GMSd AS instance. This is illustrated in figures 5.19.2.1-1 and 5.19.2.2-1 above.

2. *Centralized 5GMSd Content Preparation*. In this option, a single, primary 5GMSd AS instance encodes and packages source content that has been ingested at reference point M2d into CMMF bitstreams/objects according to a (yet to be defined) configuration provided by a Content Preparation Template(s). The CMMF bitstreams/objects created during this media processing task may be delivered directly to the 5GMSd Client (via reference point M4d), delivered to another (secondary) 5GMSd AS instance (via reference point M10), or delivered to a 5GMSd AS located in an external, possibly untrusted, Data Network. These CMMF bitstreams/objects may then be cached and/or forwarded onward. This primary 5GMSd AS instance is responsible for creation of all CMMF encoded representations used to deliver content from multiple sources. This is illustrated in figure 5.19.3.3.1-1 below. The secondary 5GMSd AS instances may be deployed either in the Trusted DN, in an Edge DN or in an External DN.



Figure 5.19.3.3.1-1: Option #2 for deploying CMMF within 5GMS
where a single, primary 5GMSd AS performs all CMMF content preparation.

3. *Decentralized 5GMSd Content Preparation.* The possibility also exists to distribute the CMMF media processing across 5GMSd Application Servers such that each 5GMSd AS instance is only responsible for creation of a single CMMF representation for which it intends to cache and/or deliver to a 5GMSd Client via reference point M4d. In this option, each 5GMSd AS instance provisioned with the Content Preparation Template may receive original source content or CMMF-encoded content from either the 5GMSd Application Provider at reference point M2d or from another 5GMSd AS instance at reference point M10d. This received content is then processed to create a new, unique CMMF-encoded representation which can be used in conjunction with others during a multi-source download. Similarly, an externally deployed 5GMSd AS may be configured by the 5GMSd Application Provider (by private means) to perform a similar media processing task to create its own CMMF-encoded representation. This is illustrated in figure 5.19.5.3.1-2 below.



Figure 5.19.3.3.1-2: Option #3 for deploying CMMF within 5GMS
where each 5GMSd AS instance performs independent CMMF content preparation.

A combination of options 1-3 is also possible where some aspect of all three exist within a physical realization of the network.

##### 5.19.3.3.2 Client-side implementation of CMMF within the 5GMS architecture

Implementing multi-source/CDN delivery using CMMF requires modifications to the 5GMSd Client. At a minimum, a 5GMSd Client must be able to download CMMF bitstreams/objects from multiple 5GMSd Application Server instances simultaneously and jointly decode the received bitstreams/objects. Options for implementing multi-source/CDN delivery using CMMF within the 5GMSd Client include:

1. *CMMF Client Proxy.* This option implements multi-source/CDN using CMMF within the client as a proxy between the Media Player and each 5GMSd AS. The proxy consists of a CMMF Client and a Media Server. Once the Media Session Handler of the 5GMSd Client has configured the CMMF Client via reference point CMMF-2, the Media Player may request source content via the Media Server using reference point CMMF-3. Once a request is received, the CMMF Client downloads different CMMF encoded representations of the requested content via reference point(s) CMMF-1 (this reference point is functionally equivalent to reference point M4 despite terminating on a different logical function in the 5GMSd Client), decodes the received CMMF bitstreams/objects, and replies to the Media Player with the requested source content via CMMF-3. This option is illustrated in figure 5.19.3.3.2-1.



Figure 5.19.3.3.2-1: Option #1 for integration of CMMF within the 5GMS Client where CMMF is implemented as a client proxy.

2. *CMMF decoder integrated in Media Player.* This option implements CMMF within the Media Player itself. An example is provided in figure 5.19.6.3.2-2 depicting CMMF integrated within the DASH-based 5GMSd Client specified in clause 13.2 of TS 26.512 [TS26512]. The architecture and operation of the 5GMS Client is similar to that in [TS26512] with the following exceptions:

a. *Download*: Downloads source content objects and/or CMMF bitstreams/objects from one or more 5GMSd AS instances in parallel.

b. *Request Scheduling:* Performs the same function as defined in clause 13.2 of [TS26512] with the addition of managing the concurrent requests sent over reference point M4 during the download of content encoded within CMMF.

c. *Throughput Estimation:* Estimates the throughput from each individual 5GMSd AS instance in addition to estimating the aggregated throughput from all 5GMSd AS instances.

d. *CMMF Receiver/Decoder:* Temporarily stores and jointly decodes CMMF bitstreams/objects as they are downloaded. Once decoded, the source content objects are moved to the Media Playback Management and Protection Controller. The CMMF Receiver/Decoder also provides status updates containing decode progress to each active download function for the purposes of managing/terminating in-process downloads.



Figure 5.19.3.3.2-2: Option #2 for integration of CMMF within the 5GMS Client where CMMF is integrated directly within the Media Player.

##### 5.19.3.3.3 CMMF service and client configuration within the 5GMS architecture

###### 5.19.3.3.3.0 General

CMMF service configuration is the overall responsibility of the 5GMSd Application Provider. The 5GMSd Application Provider may configure and provision resources to deliver media using CMMF across both external and trusted data networks.

###### 5.19.3.3.3.1 CMMF configuration information

Editor’s Note: Information required by 5GMS to host, prepare, and deliver CMMF encoded media is necessary to be provisioned by the 5GMSd Application Provider. This information may include details on how to encode and package source media within CMMF for a particular service, the locations and/or Application Servers CMMF encoded media will be hosted, which CMMF encoded representations are associated with which Application Servers, etc. Further study is required to define these necessary parameters, the impact including these parameters within 5GMS has on existing reference point APIs (particularly M1, M3, and M5), etc.

###### 5.19.3.3.3.2 CMMF content preparation on 5GMS Application Servers

Editor’s Note: It is anticipated that the configuration and provisioning of CMMF-encoded media on 5GMS Application Servers will follow the Content Preparation Template framework defined in TS 26.510 [TS26510]. Further study is required to assess the impact CMMF has on this framework.

###### 5.19.3.3.3.3 Configuration of external Application Servers

Editor’s Note: It is anticipated that the configuration and provisioning of CMMF encoded media on external DN 5GMSd Application Servers will be similar to that defined in clause 5.19.3.3.3.2 but conform to the collaboration scenarios provided in clauses A.4 and A.5 in TS 26.501 [TS26501].

###### 5.19.3.3.3.4 5GMSd Client configuration

Editor’s Note: Further study is required to assess the impacts including additional CMMF configuration information has on existing 5GMSd Client functions and reference point APIs (particularly M6d, M7d, and M11). This includes identifying methods to communicate the locations of CMMF-encoded media to and within the 5GMSd client, configuring the CMMF Receiver/Decoder/Client prior to accessing CMMF encoded media, etc.

### 5.19.4 High-level call flow

#### 5.19.4.1 Server-side CDN switching

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.4.2 Client-side CDN switching

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.4.3 Concurrent CDN access using CMMF

Support for multi-CDN media delivery can be realised by the following procedures:

1. CMMF-encoded media objects, and possibly original source media (e.g., MPEG-DASH or HLS media segments), are striped across multiple 5GMSd Application Servers. The 5GMSd Application Provider may make the CMMF-encoded media objects, and possibly original source media (e.g., MPEG-DASH or HLS media segments), available at reference point M2 or they may be created by the 5GMSd Application Server performing content preparation on regular media objects (e.g., MPEG-DASH or HLS media segments) that have been ingested from the 5GMSd Application Provider at reference point M2.

2. Upon initialization of a playback session, the 5GMSd Media Client’s Media Session Handler obtains relevant Service Access Information from the 5GMSd Application Function at reference point M5. At a minimum, this includes details concerning the location of each 5GMSd Application Server from which the CMMF-encoded and possibly original media (e.g., MPEG-DASH or HLS media segments) may be obtained, as well as appropriate signalling to indicate whether the media at each location is CMMF-encoded.

3. The 5GMSd Media Client connects to and downloads CMMF-encoded media objects, and possibly the original source media (e.g., MPEG-DASH or HLS media segments), from each 5GMSd Application Server simultaneously via reference point M4, terminating the download from each 5GMSd AS early upon obtaining enough CMMF-encoded objects to recover the source media (e.g., MPEG‑DASH or HLS media segment). Once decoded, the source media is delivered to the Media Player in the 5GMSd Media Client for presentation.

### 5.19.5 Gap analysis and requirements

#### 5.19.5.1 Server-side CDN switching

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.5.2 Client-side CDN switching

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.5.3 Concurrent CDN access using CMMF

### 5.19.6 Candidate solutions

#### 5.19.6.1 Server-side CDN switching

These candidate solutions include approaches where a media streaming client or population of clients changes or switches between two or more CDNs based on recommendations from a remote server. An example includes the DASH Industry Forum’s content steering architecture [DIFCS].

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.6.2 Client-side CDN switching

These candidate solutions include approaches where a media streaming client changes or switches between two or more CDNs based on decisions made locally.

Editor’s Note: Inclusion and expansion on these sets of solutions is dependent on interest from working group.

#### 5.19.6.3 Concurrent CDN access using CMMF

This candidate solution includes approaches where a 5GMSd Client accesses and downloads, via reference point M4, CMMF-encoded media objects [CMMF], and possibly original source media (e.g., MPEG-DASH or HLS media segments), from two or more 5GMSd Application Servers simultaneously. Additionally, the 5GMSd Client may access these 5GMSd Application Servers over different access networks (such as 3GPP and non-3GPP access networks).

### 5.19.7 Summary and conclusions

## END OF CHANGES