## **3GPP TSG SA WG4 #129-eS4-241609**

**Online, August 19 - 23 2024** *revision of* [S4aI240114](javascript:OpenContributionDetailsPopup('https://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=1581707%27,%20%27S4aI240114%27);)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *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: client side CDN switching | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Huawei, Hisilicon | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_AMD | | | | |  | ***Date:*** | | | 2024-08-13 |
|  |  | | | |  | |  | | |  |
| ***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 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-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 5.19.6.2 on client side CDN switching | | | | | | | | |
|  | |  | | | | | | | | |
| ***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 scenarios  S4-240844: updates with comments received at SA4#127-e-bis  S4aI240052: 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.  S4 S4-24160 client side CDN switching using BaseURL mechanism | | | | | | | | |

## 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)".

[TS103285] Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISO BMFF

Based DVB Services over IP Based Networks

## SECOND CHANGE

## 5.19 Multi-CDN media delivery

### 5.19.1 Description

#### 5.19.1.1 Introduction

Media streaming applications traditionally 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.

### 5.19.2 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.

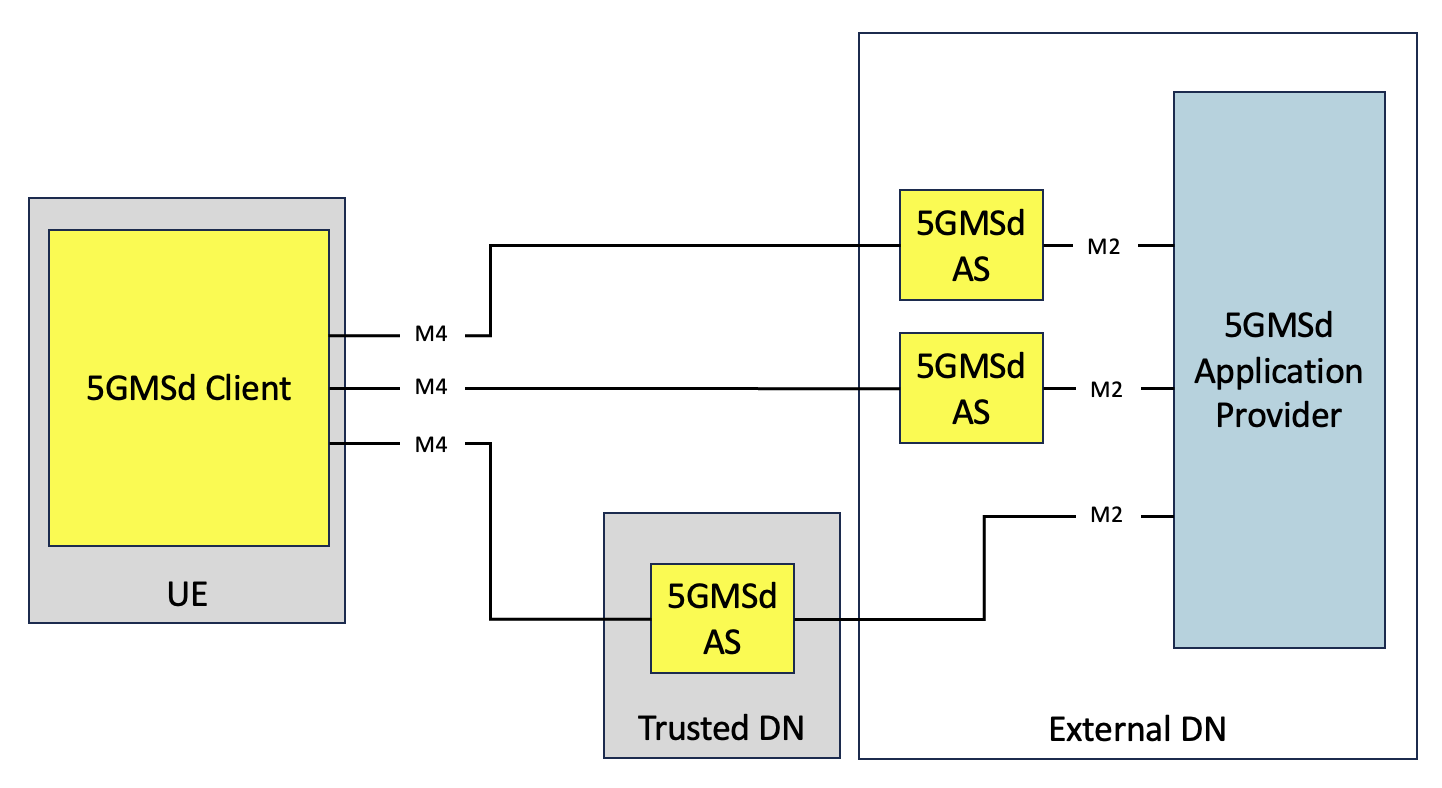


Figure 5.19.2.1-1: Multi-CDN media delivery within 5G system

#### 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.

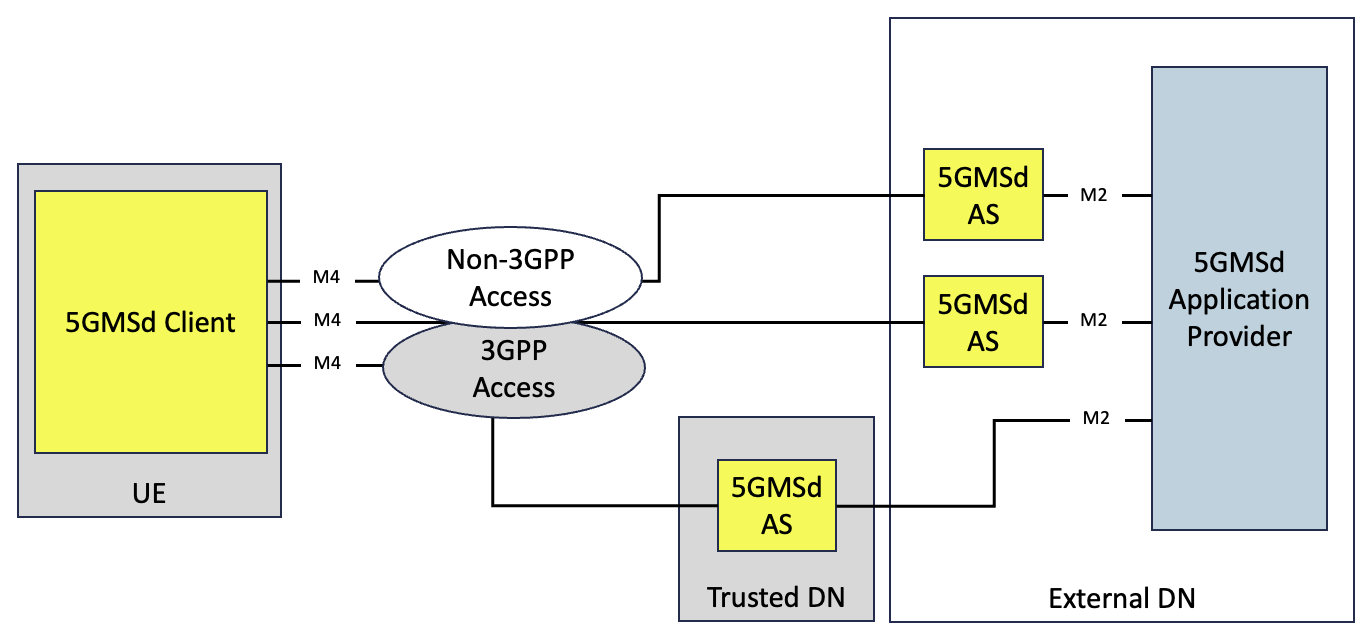


Figure 5.19.2.2-1: Multi-access media delivery within 5G system

### 5.19.3 Architecture mapping

### 5.19.4 High-level call flow

### 5.19.5 Gap analysis and requirements

### 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

##### 5.19.6.2.1 Overview

These candidate solutions include approaches where a media streaming client changes or switches between two or more CDNs based on decisions made locally. Issues include errors and load balancing in case the request load becomes large. These issues can be addressed through the use of multiple server locations and appropriate Player responses to error conditions enabling client side CDN switching.

One way to achieve this is by using multiple BaseURL in DASH MPD, this enables the client to select segments from different servers. The solution presented here is based on the solution in ETSI TS 103 285 DVB-DASH [TS103285] and uses MPEG-DASH. Other similar ways of using the MPEG-DASH BaseURL may also be considered.

##### 5.19.6.2.2 Handling of BaseURLs by Players

According to ISO/IEC 23009-1, MPDs may contain BaseURLs at many levels (within the MPD, Period, AdaptationSet and Representation elements). At each of these levels there can be more than one BaseURL. The presence of multiple BaseURLs at any given level indicates that the same content can be obtained from more than one location. This enables to increase resilience and provide load balancing by offering different distribution routes - for example different CDNs.

Where players find, having evaluated the rules as specified in ISO/IEC 23009-1 clause 5.6 for resolving relative URLs (if necessary), that multiple BaseURLs are present for an item then they can make use of the alternatives offered in the event of an error with the first one used, as described in this clause.

A new attributeGroup, containing two attributes may be defined to finetune the BaseURL functionality and enable client side switching. The main idea is to introduce two additional attributes:

The baseUrlAttributes group is defined as follows:

<xs:attributeGroup name="baseUrlAttributes">

<xs:attribute name="priority" type="xs:positiveInteger" use="optional" default="1"/>

<xs:attribute name="weight" type="xs:positiveInteger" use="optional" default="1"/>

</xs:attributeGroup>

The two attributes from the baseUrlAttributes group are used together with the serviceLocation attribute on BaseURL elements to provide a mechanism by which content providers can signal to Players the priority, weight and independence of different BaseURLs. The attributes are defined as:

* @priority is a positive integer. It has a default value of 1. Lower values indicate higher priority levels.
* @weight is a positive integer. It has a default value of 1. The weight describes the relative weighting of BaseURLs with the same priority, with the semantics that a BaseURL with a weight of 2 is twice as likely to be chosen by a Player as one with a weight of 1. This is used to indicate how the content provider needs traffic to be distributed between BaseURLs of a given priority and is usually related to the relative capacity of the infrastructure serving the different BaseURLs.
* @serviceLocation is an optional string described further below. When @serviceLocation is not set on an absolute BaseURL then serviceLocation can take the value of the BaseURL (that is the serviceLocation is the URL inside the element).

Players can carry out BaseURL reference resolution as specified in clause 5.6.4 of ISO/IEC 23009-1 with the following caveats relating to the additional attributes:

* The document base URI is treated as if it were a BaseURL element with the default values for the @priority and @weight attributes, and a @serviceLocation value of the document base URI.
* Players can associate values of @priority, @weight and @serviceLocation attributes with resolved BaseURL elements. These can be taken from the BaseURL element that provided an absolute URL for the resolution process, or the document base URI if that was used as the absolute URL. Hence, Players can ignore any @priority, @weight or @serviceLocation attributes that are present on BaseURL elements containing relative URLs.

The @serviceLocation attribute can in addition be used to implement a blacklisting of failed BaseURL locations. To do this the Player can maintain a list of @serviceLocation values which have failed and returned error responses. When an MPD is first loaded in a session the blacklist is empty. The blacklist is retained when the MPD is reloaded by the Player, but discarded when a different MPD is loaded or at the end of the current session.

Whenever the Player needs to construct a list of BaseURLs, all URLs in the list which have a @serviceLocation attribute matching an entry in the blacklist can be removed from the available list of BaseURLs. Additionally, whenever a BaseURL is removed from the available list of BaseURLs, any other BaseURL with the same @priority value as the BaseURL being removed can also be removed.

This reduced list of BaseURLs is used when selecting a BaseURL as described in the following clauses.

When a Player needs to use a BaseURL to resolve a reference to external content, such as may be found inside a Segment Template, it picks the BaseURL as follows:

* It begins by taking the set of resolved BaseURLs present or inherited at the current position in the MPD, that have the lowest @priority attribute value.
* If there is more than one BaseURL with this lowest @priority attribute value then the Player selects one of them at random such that the probability of each BaseURL being chosen is proportional to the value of its @weight attribute.
* If there are no BaseURLs after applying blacklisting, the Player can stop playback and report an error.

Once a random selection has been carried out amongst a group of BaseURLs with the same @priority attribute value, then that choice can be re-used if the selection needs to be made again unless the blacklist has been modified or the available BaseURLs have changed.

At any point where a Player needs to change BaseURL the Player can:

* Add the @serviceLocation attribute value to the blacklist. This BaseURL is removed from the list of available BaseURLs, as are any other BaseURLs in the list with the same @priority attribute value.
* Select the set of BaseURLs from the list available BaseURLs that have the lowest @priority attribute value.
* If there is more than one BaseURL in this set, use the @weight attribute, to select between them.
* If a BaseURL with the same @serviceLocation attribute is in use elsewhere by the Player (for example in accessing content for a different Adaptation Set) then that BaseURL can be replaced following the rules given in this clause when the next request for media is to be made.

##### 5.19.6.2.3 BaseURL Example

Take the following MPD excerpt as an example:



Initially the Player has an empty serviceLocation blacklist.

When the Player needs to access the media within period "p1" it will resolve the BaseURL within the period element and end up with the following set of BaseURLs:



Note that the BaseURL with serviceLocation "D" has been assigned the default weight of "1". No BaseURLs are excluded since the serviceLocation blacklist is empty.

From this it takes the BaseURL(s) with the lowest priority:



It then needs to pick one of them to use. It adds up the weight values of all these BaseURLs which gives a value of 100. It then picks a random integer in the range from 0 to 99 (inclusive). It then selects the BaseURL using that number (called rn here) as follows:

* 0 ≤ rn<10 🡪 http://cdn1.example.com/period/
* 10 ≤ rn<40 🡪 http://cdn2.example.com/period/
* 40 ≤ rn<100 🡪 http://cdn3.example.com/period/

In this case consider the random number it picked is 30, which gives the value of rn. The BaseURL picked is "http://cdn2.example.com/period/" with @serviceLocation "B".

It would then access the initialization segment for the Adaptation Set shown in the example as "http://cdn2.example.com/period/rep1/IS" and the media segments as "http://cdn2.example.com/period/rep1/$Number$".

Now consider a fault occurs which triggers the Change URL The blacklist of serviceLocations are now updated to include "B".

Upon examining the list of BaseURLs the Player now discards any BaseURL with a serviceLocation of "B" or with a priority of "1".



Now the lowest priority which it can use is 3, so the BaseURL(s) with that priority are selected:



There is only one BaseURL, so the random number selection can be skipped. The BaseURL value is now "http://cdn3x.example.com/period/".

Now, consider again a condition requiring the Change BaseURL behaviour. This leads to "C" being added to the serviceLocation blacklist. That now contains "B" and "C".



The Player now finds the lowest priority value is 5 and selects the BaseURLs with that value:



As there are two, the Player will pick at random again. It adds up the weight values of all these BaseURLs which gives a value of 2. It then picks a random integer between 0 and 1 (inclusive). It then selects the BaseURL using that number (called rn here) as follows:

* 0 ≤ rn<1 🡪 http://cdn4.example.com/period/
* 1 ≤ rn<2 🡪 http://cdn5.example.com/example/period/

In this case consider the random number it picked is 1. The BaseURL picked is "http://cdn5.example.com/example/period/" with serviceLocation "E".

Finally consider a further fault requiring the Change BaseURL behaviour. The value "E" is now added to the serviceLocation blacklist and all BaseURLs with serviceLocation of "E" or priority of "5" are removed from the list of available BaseURLs. The list of the available BaseURLs now contains:



There are no more available so the Player reports an error to the application controlling it and ends the session.

##### 5.19.6.2.4 DNS - HTTP Player

DNS records for a given hostname may provide more than one address. Players are able to make use of at least one additional address in the event that the first address received in the response is unreachable or refuses connections.

##### 5.19.6.2.5 Types of error condition and recovery options

There are a number of types of error condition which may lead to problems. It may not necessarily be obvious which has occurred, but by grouping them into categories a strategy for dealing with each category can be used.

Network congestion in the home or the Player's ISP is not an error condition and also is something which adaptive streaming is designed to work around. However congestion at the server end, or servers suffering from high loading, may be avoidable if alternative servers with more available capacity can be used.

Configuration errors may appear when MPDs incorrectly contain links to CDNs which have not been configured to serve the content within the MPD or which have lost connectivity with the content provider. Although this is a situation which can normally not happen, the possibility needs to be taken into account.

Authentication errors could occur for these reasons:

* By mistake - for example as a configuration error.
* Intentionally - for example a GeoIP check failing.
* A time limited token having expired.

Three problems are collated here as missing segments, however the causes are significantly different and so the response to such errors needs care to avoid causing additional problems. Problems which may lead to segments being irretrievable are:

* Equipment failure at the content provider, leading to media segments from one content production path being unavailable. A Player may be able to recover from this by trying alternate BaseURLs as other paths may still be operating.
* Player/server time misalignment when playing a live stream. This may cause the Player to attempt to retrieve segments before they become available on the server, or after they have left the timeshift buffer.
* Intermittent faults in the content provider or distribution network causing occasional segments to be unavailable.

There are some HTTP statuses which indicate an error, but are not expected to be encountered. If they are encountered and the Player is unaware of why it has happened then they can be treated in the same way as a configuration error.

This category covers errors which lead to dropped or stalled connections, without an identified cause and which do not recur if the connection is reopened. It also covers erratic data transfer caused by congestion.

This clause lists errors as they are likely to be seen at a Player and the category they belong in. Table XX gives some of the categories of errors as defined in [TS103285]

Table XX: Specific errors and their categories

| Connection | HTTP Status Code | Error category | Notes |
| --- | --- | --- | --- |
| DNS resolution failed | N/A | Heavy server load | Although this may be caused by something other than heavy server load, the actions to take in response to this are the same |
| Host unreachable | N/A | Heavy server load |  |
| Connection refused | N/A | Heavy server load |  |
| Connection or packet transfer ('socket') timeout | N/A | Transient connection problems or congestion | These are problems which are likely to be caused by network errors, possibly close to the Player, which may disappear if the connection retried |
| Congestion related problems | N/A | Transient connection problems or congestion | For example unexpectedly slow or bursty connections |
| Connection closed before response complete | 200, 206 | Miscellaneous request errors | Possible transient error in CDN |
| OK | 200, 206 but resulting in segment parse error, e.g. truncation | Miscellaneous request errors | Possible transient error in CDN |
| OK | 401 (Unauthorized) | Authentication errors |  |
| OK | 402, 403 | Authentication errors | Might indicate a token is invalid |
| OK | 404 (Not found) | Missing segments |  |
| OK | 405 (Method not allowed)  406 (Not acceptable)  407 (Proxy authentication required)  409 (Conflict)  411 (Length required)  412 (Precondition failed)  413 (Request entity too large)  414 (Request-URI too long)  415 (Unsupported media type)  417 (Expectation failed) | Miscellaneous request errors | These can normally not really occur in this use |
| OK | 408 (Request timeout) | Miscellaneous request errors | Although this is a retry-able error, it may be the Player is not sending the whole request |
| OK | 410 (Gone) | Missing segments | Might be used on a live stream to indicate something has been deleted, but this is not guaranteed |
| OK | 416 (Requested range not satisfiable) | Missing segments |  |
| OK | 500 (Internal server error) | Heavy server load |  |
| OK | 501 (Not implemented) | Miscellaneous request errors | These can normally not really occur in this use |
| OK | 502 (Bad gateway) | Configuration errors |  |
| OK | 503 (Service unavailable) | Heavy server load |  |
| OK | 504 (Gateway timeout) | Heavy server load |  |
| OK | 505 (HTTP version not supported) | Miscellaneous request errors | These can normally not really occur in this use |

##### 5.19.6.2.5 Requirements on Players

Where Players encounter an error identified in Table XX then they need to take action to try to enable them to continue.

This clause gives some requirements to ensure that failover between BaseURLs happens in the event of a failure, and to avoid flooding content servers with requests in the event of a configuration error or complete stream failure.

Players can follow the action specified in the "Action to take" column of when they encounter an error, from the row with the appropriate values in the "Error category" and "MPD@type" columns of that table. Where an action specifies "retry" the Player can close and reopen the connection to the server before making the HTTP request. Players can use a suitable delay between retries, balancing normal backoff etiquette with the need to avoid disrupting playback. Where an action specifies "change BaseURL", there is no requirement for the Player to remember previous retries after changing BaseURL.

Table 29: Action to take in reaction to errors in the different categories

| Error Category | MPD@type | Action to take | Maximum Number of retries |
| --- | --- | --- | --- |
| Heavy server load | static or dynamic | The Player may retry the request up to the max number or retries specified. If the problem persists it can change BaseURL. | 1 |
| Missing segments | static | The Player can change BaseURL. |  |
| Missing segments | dynamic | The Player can reload the MPD and resynchronize.  If, as a result of reloading the MPD and performing any required time synchronization, the Player determines the request is no longer appropriate, it shall adjust its position in the media to reflect the new MPD and any new time value.  If the request is still valid the Player may retry the request up to the max number of retries specified.  If trying the above does not result in success the Player can change BaseURL | 2 |
| Configuration errors | static or dynamic | The Player may retry the request up to the max number of retries specified. If the problem persists it can change BaseURL | 1 |
| Miscellaneous request errors | static or dynamic | The Player may retry the request up to the max number of retries specified. If the problem persists it can change BaseURL | 1 |
| Authentication errors | static or dynamic | The Player may retry the request up to the max number of retries specified. If the problem persists it can change BaseURL | 1 |
| Transient connection problems or congestion | static or dynamic | Players can attempt to deal with these problems through the adaptive bitrate system, retrying requests where connections time out and changing bitrate if congestion causes poor throughput. |  |

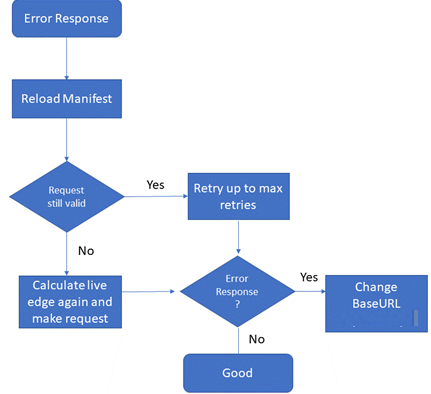


Figure XX: Handling missing segments in case of more than one BaseURL

##### 5.19.6.2.6 Dynamically changing adding or removing of BaseURLs

In case of a n MPD update as used for MPD@type=”dynamic” BaseURLs can be added or removed, and weights can be modified. This allows some level of influence on the player/clients decisioning by the content generator.

##### 5.19.6.2.7 Discussion

The solution uses exising known and adopted technologies. It would require mostly some updates on player implementation streaming client implementation.

#### 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).

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.

Generally, support for multi-CDN and/or multi-access media delivery using CMMF can be realized by the following:

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. Detailed examples for preparing original source media for delivery from multiple serving endpoints using CMMF are provided in [CMMF].

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.

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

##### 5.19.6.3.1 Network-side implementation of CMMF

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 an external, possibly untrusted, Application Server. 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 5.19.6.3.1-1 below. The secondary 5GMSd AS instances may be deployed either in the Trusted DN, in an Edge DN or in an External DN.

3. *Decentralized 5GMS Network Media Processing.* 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, possibly untrusted, Application Server 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 5.19.6.3.1-2 below.

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

A diagram of a computer

Description automatically generated

Figure 5.19.6.3.1-1: Option #2 for deploying CMMF within 5GMS  
where a single, primary 5GMSd AS performs all CMMF content preparation.

A diagram of a computer system

Description automatically generated

Figure 5.19.6.3.1-2: Option #3 for deploying CMMF within 5GMS  
where each 5GMSd AS instance performs independent CMMF content preparation.

##### 5.19.6.3.2 Client-side implementation of CMMF

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 5.19.6.3.2-1.

A diagram of a computer application

Description automatically generated

Figure 5.19.6.3.2-1: 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.

#### A yellow diagram with many different symbols Description automatically generated with medium confidence

Figure 5.19.6.3.2-2: Option #2 for integration of CMMF within the 5GMS Client where CMMF is integrated directly within the Media Player.

##### 5.19.6.3.3 CMMF service and client configuration

###### 5.19.6.3.3.0 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.

###### 5.19.6.3.3.1 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.

###### 5.19.6.3.3.2 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.

###### 5.19.6.3.3.3 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 5.19.6.3.3.2 but conform to the collaboration scenarios provided in A.4 and A.5 in TS 26.501 [TS26501].

###### 5.19.6.3.3.4 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.

### 5.19.7 Summary and conclusions

## END OF CHANGES