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

**Online, August 19 - 23 2024** *revision of* S4aI240114

|  |
| --- |
| *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 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 5.19.6.2 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 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.S4 S4-24160 client side CDN switching using BaseURL mechanism |

## FIRST CHANGE

## 2 References

[TS103285] ETSI 103 285: "Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks".

## SECOND CHANGE

### 5.19.1 Description

#### 5.19.1.3 Handling of multiple BaseURLs by MPEG-DASH 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 playback resilience to be improved and to provide load balancing by offering different distribution routes, for example different CDNs.

Where Players find, having evaluated the rules as specified in clause 5.6 ISO/IEC 23009-1 [?] 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 later in this Key Issue.

## THIRD CHANGE

### 5.19.3 Architecture mapping

#### 5.19.3.2 Client-side CDN switching

Editor’s Note: First state which collaboration scenarios form clause 5.19.2 or annex A of TS 26.501 are relevant. Is multi-access relevant or just multi-CDN?

## FOURTH CHANGE

### 5.19.4 High-level call flow

#### 5.19.4.2 Client-side CDN switching

## FIFTH CHANGE

### 5.19.5 Gap analysis and requirements

#### 5.19.5.2 Client-side CDN switching

## SIXTH CHANGE

### 5.19.6 Candidate solutions

#### 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 BaseURLs in the DASH MPD. This enables the client to select segments from different servers. The solution presented here is based on the solution in TS 103 185 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 Fine-tuned selection of BaseURL

A new attributeGroup, containing two attributes may be defined to fine-tune 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.

## END OF CHANGES