**3GPP TSG- #130**

**, United States, -** *revision of S4aI240198*

|  |
| --- |
| *CR-Form-v12.3* |
| **PSEUDO CHANGE REQUEST** |
|  |
|  |  | **CR** |  | **rev** |  | **Current version:** |  |  |
|  |
| *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 |  |

|  |
| --- |
|  |
| ***Title:***  |  |
|  |  |
| ***Source to WG:*** | Xiaomi, BBC |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** |  |  | ***Date:*** |  |
|  |  |  |  |  |
| ***Category:*** |  |  | ***Release:*** |  |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories 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:*** | No high-level call flows yet in the QUIC-based segment delivery in 5GMS clause. |
|  |  |
| ***Summary of change:*** | Based on the discussion on S4aI240165, representing call flows in a generalized manner, and thus abstracted from a concrete application layer protocol, creates ambiguities and confusion. proposed Therefore, this contribution (revision of S4aI240198) aims at proposing a solution to address this issue by:1. Choosing one application layer protocol as example protocol for each 5GMS Client type’s call flow.
2. Changing the call flows to correspond to the example application layer protocol.

This would keep the future conclusions of the WT application-layer-agnostic while giving example instantiations of the 5GMS client types. |
|  |  |
| ***Consequences if not approved:*** | No progress of the QUIC topic in FS\_AMD. |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **x** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **x** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

Change #1 (new text compared to CR 19)

### 5.24.4 High-level call flows

#### 5.24.4.1 General

This clause collects high-level call flows highlighting steps where the usage of QUIC or a QUIC-based protocol for the delivery of segmented content would have an impact on the 5GMS System.

#### 5.24.3.2 Impact of a QUIC-agnostic 5GMS Client on the 5GMS high-level procedure for DASH content

For the purpose of illustration, and to assist later gap analysis, an HTTP/3 client instantiated inside a Media Player provides a concrete example of an application layer protocol used in the context of a QUIC-agnostic 5GMS Client. In this case, the DASH MPD and the segments are requested and served within a QUIC connection underlying the HTTP/3 session. The usage of QUIC as transport protocol in this case has no impact on the steps of the high-level procedure for DASH content defined in clause 5.2 of TS 26.501 [15]. The same 19 steps are presented. However, the following notes describe the differences which are blue-highlighted in the sequence diagram in figure 5.24.3.2-1.

![Msc-generator~|version=8.6.1~|lang=signalling~|size=874x921~|text=numbering=yes;~nhscale=auto;~n~nApp[label=~q5GMSd-Aware \nApplication~q];~nplayer[label=~qMedia\nPlayer~q];~nsessionHnd[label=~qMedia\nSession\nHandler~q];~naf[label=~q5GMSd\nAF~q];~nserver[label=~q5GMSd\nAS~q];~next[label=~q5GMSd \nApplication \nProvider~q];~n~ndefstyle quic [text.color=blue, text.bold=true, line.color=blue, arrow.color=blue];~n~n~nvspace 10;~nApp--ext: Service Announcement and Content Discovery {~n~5App~gext: Get Media Content Info[number=no];~n~5ext~gApp: List of Media Content Descriptions\n\-(List of Entry URls with additional metadata)[number=no];~n};~nvspace 5;~nApp--App: Select\nMedia Content;~nApp-~gplayer-~gsessionHnd: Initiate Media Playback\n\-(Media Player Entry);~n..: [tag=~qopt~q] {~n~4sessionHnd~l-~gaf [arrow.type=dot]: Service Access Information\nacquisition;~n};~nhide sessionHnd, af;~n~nApp-~gplayer: Start Media Playback\n\-(Media Player Entry);~nhide App;~nplayer~l-~gserver [arrow.type=dot, quic]: Establish transport session for the manifest;~nplayer-~gserver [quic]: Request MPD (Entry Point);~nserver-~gplayer [quic]: OK\n\-(MPD);~n~nshow sessionHnd;~nplayer--player: Process MPD;~nplayer-~gsessionHnd: MPD Rx Notification;~n~nvspace 5;~nplayer..ext: [tag=~qopt~q]{~n~4player~l-~gext [arrow.type=dot]: DRM License aquisition;~n};~nhide ext;~n~nvspace 5;~nplayer--player: Configure playback\npipeline;~n~nvspace 5;~nplayer~l-~gserver [arrow.type=dot, quic]: Establish transport session for content\n\-(optional Transport Session Parameters);~nplayer-~gsessionHnd: Notification\n\-(Transport Session Parameters);~nhide sessionHnd;~n~nvspace 5;~n-- [tag=~qloop~q]: {~n~4player-~gserver [quic]: Request Initialization Information(s);~n~4server-~gplayer [quic]: OK\n\-(Initialization Information(s));~n};~n~nvspace 5;~n-- [tag=~qloop~q, number=19]: \IRepeat {~n~4player-~gserver [number=17, quic]: Request Media Segment(s);~n~4server-~gplayer [quic]: Media Content;~n};~n~|gui_state=daakdaakdaakdbakdccadbakfmabcidbcmdbdfcmdecmdacjfmabcidbcmdbdfcmdgcmdacjakfdgfhchggjgdgfcaebgogogphfgogdgfgngfgohecagbgogecaedgpgohegfgohecaeegjhdgdgphggfhchjakebhahaakgfhiheakdcakdaakdaak~|]()

Figure 5.24.3.2-1 High-level call flow for DASH content using a QUIC-agnostic 5GMS Client
based on figure 5.2‑2 from TS 26.501 [15]

The specifics of using QUIC-based protocol in this procedure are:

6: The Media Player establishes a **QUIC** connection with the 5GMSd Application Server.

7: The MPD is requested using an **HTTP/3** request message **which goes over a new client-initiated QUIC stream** in the **QUIC** connection established in step 6.

8: The MPD is delivered **over the QUIC stream** initiated in step 7.

13: **If the QUIC connection of step 6 is no longer open, the QUIC Client establishes a new connection with the QUIC server.**

15: The Initialization Segments are requested using an **HTTP/3** request message **which goes over a new client-initiated QUIC stream in the open QUIC connection**.

16: The Initialization Segment response is delivered **over the QUIC stream initiated in step 15**.

17: The media segment is requested using an **HTTP/3** request message **which goes over a new client-initiated QUIC stream in the open QUIC connection**.

18: The media segment response is delivered **over the QUIC stream initiated in step 17**.

NOTE: After step 18, the Media Player may have to establish a new connection whenever necessary. This can, for instance, be the case if the 5GMSd AS closes the QUIC connection due to inactivity by the Media Player.

#### 5.24.3.3 Impact of a Media-independent QUIC-aware 5GMS Client on the 5GMS high-level procedure for DASH content

For the purpose of illustration, and to assist later gap analysis, a HTTP/3 client instantiated inside a Media Player provides a concrete example of an application layer protocol used in the context of a Media-independent QUIC-aware 5GMS Client. In this context, the DASH MPD and the segments are typically requested and served within the same QUIC connection which is kept alive by the 5GMS Client.

Compared with the 5GMS high-level procedure for DASH content defined in clause 5.2 of TS 26.501 [15], the possible backward compatible enhancements are:

- A new step 8a allows the 5GMSd AS to optionally push Initialization Segments to the Media Player using HTTP/3 server push.

- Steps 15 and 16 are omitted unless the Media Player requires different Initialization Segments than those pushed in step 8a, or if step 8a has not been performed.

- In step 17, the request for each media segment by the Media Player may be enhanced with the ability to indicate a request priority.

However, the following notes can be made and are captured with blue highlight in the sequence diagram in figure 5.24.3.3-1.

![Msc-generator~|version=8.6.1~|lang=signalling~|size=879x1162~|text=numbering=yes;~nhscale=auto;~n~nApp[label=~q5GMSd-Aware \nApplication~q];~nplayer[label=~qMedia\nPlayer~q];~nsessionHnd[label=~qMedia\nSession\nHandler~q];~naf[label=~q5GMSd\nAF~q];~nserver[label=~q5GMSd\nAS~q];~next[label=~q5GMSd\nApplication\nProvider~q];~n~ndefstyle quic [text.color=blue, line.color=blue, line.color=blue, arrow.color=blue, text.bold=yes, tag.text.color=blue, tag.line.color=blue];~n~n~nvspace 10;~nApp--ext: Service Announcement and Content Discovery {~n~4App~gext: Get Media Content Info[number=no];~n~4ext~gApp: List of Media Content Descriptions\n\-(List of Entry URls with additional metadata)[number=no];~n};~nvspace 5;~nApp--App: Select\nMedia Content;~nApp-~gplayer-~gsessionHnd: Initiate Media Playback\n\-(Media Player Entry);~nsessionHnd..af: [tag=~qopt~q]{~n~4sessionHnd~l-~gaf [arrow.type=dot]: Service Access Information\nacquisition;~n};~nhide sessionHnd, af;~n~nApp-~gplayer: Start Media Playback\n\-(Media Player Entry);~nhide App;~n~nplayer~l-~gserver [arrow.type=dot]: Establish transport session for the manifest;~n~nvspace 5;~n-- [number=no, quic]: \IQUIC Stream {~n~4player-~gserver: Request MPD (Entry Point);~n~4server-~gplayer: OK\n\-(MPD);~n~4vspace 5;~n~4player..server: [tag=~qopt~q, quic]{~n~8server-~gplayer [number=no, quic]: 8a: Pushed Initialization Information;~n~8player--player [number=no, quic]: 8b: Store Initialization Information\nin HTTP client cache;~n~4};~n};~n~nvspace 5;~nshow sessionHnd;~nplayer--player: Process MPD;~nplayer-~gsessionHnd: MPD Rx Notification;~n~nvspace 5;~nplayer..ext: [tag=~qopt~q]{~n~4player~l-~gext [arrow.type=dot]: DRM License aquisition;~n};~nhide ext;~n~nvspace 5;~nplayer--player: Configure\nplayback pipeline;~n~nplayer~l-~gserver [arrow.type=dot]: Establish transport session for content\n\-(optional Transport Session Parameters);~n~nplayer-~gsessionHnd: Notification\n\-(Transport Session Parameters);~nhide sessionHnd;~n~nvspace 5;~nbox .. [tag=~qopt~q, number=no, quic]: \IInitialization Information not already in HTTP client cache per step 8b {~n~4-- [tag=~qloop~q, number=no]: {~n~8-- [number=no, quic]: \IQUIC Stream {~n~9~3numbering=yes;~n~9~3player-~gserver: Request Initialization Information;~n~9~3server-~gplayer: OK\n\-(Initialization Information);~n~8};~n~4};~n};~n~nvspace 5;~n-- [tag=~qloop~q, number=19]: \IRepeat {~n~4-- [number=no, quic]: \IQUIC Stream {~n~8player-~gserver [number=19, quic]: Request Media Segment;~n~8server-~gplayer: Media Content;~n~4};~n};~n~|gui_state=daakdaakdaakdbakdccadbakfmabcidbcmdbdfcmdecmdacjfmabcidbcmdbdfcmdgcmdacjakfdgfhchggjgdgfcaebgogogphfgogdgfgngfgohecagbgogecaedgpgohegfgohecaeegjhdgdgphggfhchjakebhahaakgfhiheakdcakdaakdaak~|]()

Figure 5.24.3.3-1 High-level call flow for DASH content using an HTTP/3 client as a
media-independent QUIC-aware 5GMS Client

The specifics of using an HTTP/3 client as media-independent QUIC-aware 5GMS Client in this procedure are:

**8a: The 5GMSd AS may send Initialization Segments to the HTTP client cache of the Media Player ahead of time using HTTP/3 server push functionality. Each Initialization Segment is pushed down the same QUIC stream as used for the MPD request in step 7 and hence the same QUIC connection established in step 6. (The QUIC connection can remain open until segment requests are subsequently sent by the Media Player.)**

**8b: Pushed Initialization Segments are cached by the Media Player.**

15: **If not already pushed to and cached by the Media Player in step 8b,** the Media Player requests Initialization Segments from the 5GMSd AS. **Each segment request/response consumes a QUIC client-initiated stream in the HTTP/3 protocol mapping.**

17: The Media Player sends requests for media segments. **Each segment request/response consumes a QUIC client-initiated stream in the HTTP/3 protocol mapping.** **If sent concurrently, the Media Player may send priority information along with the media segment requests using the Extensible Prioritization Scheme specified in RFC 9218 [HTTP-PRIO], which is supported by HTTP/3.**

18: The 5GMSd AS responds to the requests **according to the prioritisation**.

Regarding step 8a, the benefit is to decrease the media presentation start-up time by avoiding the need for the Media Player to make additional requests for initialization segments. In this case, the Media Player performs step 16a instead of steps 15 and 16.

For steps 17 and after, a Media Player may send several requests for segments of the same Representation, of different Adaptation Sets, etc. In this case, the Media Player is able to express relative priorities and expects quicker reception of the ones that matter most, for instance the next segment on the media timeline compared to future ones, audio segments compared to video segments, etc. Those possibilities are listed in clause 5.24.2.3.

NOTE: Prioritisation by itself is not supported by the QUIC protocol natively but enabled by prioritising QUIC streams when multiplexing on the UDP connection. Also, HTTP/3 support the Extensible Prioritization Scheme specified in RFC 9218 [HTTP-PRIO] which allows an HTTP/3 client to express request priorities. These techniques are described in clause 5.24.1.4.

#### 5.24.3.4 Impact of a media-optimised QUIC-aware 5GMS Client on the 5GMS high-level procedure for DASH content

For the purpose of illustration, and to assist later impact analysis, a client sharing some commonalities with a Media-over-QUIC client instantiated inside a Media Player provides a concrete example of an application layer protocol used in the context of a Media-optimised QUIC-aware 5GMS Client. However, this analysis deviates from the [MoQ] specification by assuming the presence of an MPD for content selection and the presence of explicit client segment requests instead of a subscription-based mechanism. What remains common is the mapping of media segment onto QUIC streams. In this context, the DASH MPD and the media segments are also requested and served within the same QUIC connection which is kept alive by the Media Player.

Compared with the call flow for a media-independent QUIC-aware 5GMS Client presented in clause 5.24.3.3, the differences are:

- Requests for media segments in steps 17 and 18 are grouped over the same QUIC stream.

- There can be one or more repetitions of the steps 17i and 18i within each QUIC stream over time.

The concept of groups aims at encompassing different approaches when using a media-optimised application layer protocol of which [MoQ] is one example:

- A first approach is to use a single group for all media segment requests, which has the effect of treating all segment requests equally on a first-in-first-out basis.

- Another approach is to group media segment requests per DASH Representation, i.e. one group per quality of audio and video components.

- Yet another approach is to group segment requests per Adaptation Set.

- Finally another approach is to use one group per segment request. This approach corresponds to the HTTP/3 case by definition of the HTTP/3 specification in RFC 9114 [5].

Those differences compared with the 5GMS high-level procedure for DASH content are blue-highlighted in the sequence diagram in figure 5.24.3.4-1.

![Msc-generator~|version=8.6.1~|lang=signalling~|size=879x1252~|text=numbering=yes;~nhscale=auto;~n~nApp[label=~q5GMSd-Aware \nApplication~q];~nplayer[label=~qMedia\nPlayer~q];~nsessionHnd[label=~qMedia\nSession\nHandler~q];~naf[label=~q5GMSd\nAF~q];~nserver[label=~q5GMSd\nAS~q];~next[label=~q5GMSd\nApplication\nProvider~q];~n~ndefstyle quic [text.color=blue, line.color=blue, line.color=blue, arrow.color=blue, text.bold=yes, tag.text.color=blue, tag.line.color=blue];~n~n~nvspace 10;~nApp--ext: Service Announcement and Content Discovery {~n~5App~gext: Get Media Content Info[number=no];~n~5ext~gApp: List of Media Content Descriptions\n\-(List of Entry URls with additional metadata)[number=no];~n};~nvspace 5;~nApp--App: Select\nMedia Content;~nApp-~gplayer-~gsessionHnd: Initiate Media Playback\n\-(Media Player Entry);~nsessionHnd..af: [tag=~qopt~q]{~n~4sessionHnd~l-~gaf [arrow.type=dot]: Service Access Information\nacquisition;~n};~nhide sessionHnd, af;~n~nApp-~gplayer: Start Media Playback\n\-(Media Player Entry);~nhide App;~n~nplayer~l-~gserver [arrow.type=dot]: Establish transport session for the manifest;~n~nvspace 5;~n-- [number=no, quic]: \IQUIC Stream {~n~4player-~gserver: Request MPD (Entry Point);~n~4server-~gplayer: OK\n\-(MPD);~n~4vspace 5;~n~4player..server: [tag=~qopt~q, quic]{~n~8server-~gplayer [number=no, quic]: 8a: Pushed Initialization Information;~n~8player--player [number=no, quic]: 8b: Store Initialization Information\nin HTTP client cache;~n~4};~n};~n~nvspace 5;~nshow sessionHnd;~nplayer--player: Process MPD;~nplayer-~gsessionHnd: MPD Rx Notification;~n~nvspace 5;~nplayer..ext: [tag=~qopt~q]{~n~4player~l~gext [arrow.type=dot]: DRM License aquisition;~n};~nhide ext;~n~nvspace 5;~nplayer--player: Configure\nplayback pipeline;~nplayer~l-~gserver [arrow.type=dot]: Establish transport session for content\n\-(optional Transport Session Parameters);~nplayer-~gsessionHnd: Notification\n\-(Transport Session Parameters);~nhide sessionHnd;~n~nvspace 5;~nbox .. [tag=~qopt~q, number=no, quic]: \IInitialization Information not already in HTTP client cache per step 8b {~n~4-- [tag=~qloop~q, number=no]: {~n~8-- [number=no, quic]: \IQUIC Stream {~n~9~3numbering=yes;~n~9~3player-~gserver: Request Initialization Information;~n~9~3server-~gplayer: OK\n\-(Initialization Information);~n~8};~n~4};~n};~n~nvspace 10;~n-- [tag=~qpar~q, number=19, quic]: \IIn parallel for each group {~n~4-- [number=no, quic]: \IQUIC Stream per group {~n~8numbering.pre=~q17\_~q;~n~8numbering.post=~q\^\+: ~q;~n~8player-~gserver [number=1, quic]: Request Media Segment(s);~n~8numbering.pre=~q18\_~q;~n~8server-~gplayer [number=1, quic]: Media Content;~n~n~8numbering.pre=~q17\_~q;~n~8numbering.format=~qabc~q;~n~8numbering.post=~q\^\+: ~q;~n~8player-~gserver [number=~qn~q, quic]: Request Media Segment(s);~n~8numbering.pre=~q18\_~q;~n~8server-~gplayer [number=~qn~q, quic]: Media Content;~n~8...;~n~4};~n};~n~|gui_state=daakdaakdaakdbakdccadbakfmabcidbcmdbdfcmdecmdacjfmabcidbcmdbdfcmdgcmdacjakfdgfhchggjgdgfcaebgogogphfgogdgfgngfgohecagbgogecaedgpgohegfgohecaeegjhdgdgphggfhchjakebhahaakgfhiheakdcakdaakdaak~|]()

Figure 5.24.3.4-1 High-level call flow for DASH content using a media-optimised QUIC-aware 5GMS Client

The specifics of using QUIC in this procedure are:

**8a: The 5GMSd AS may send the Initialization Segments to the Media Player ahead of time via the same QUIC stream as used for the MPD request in step 7 and hence the same QUIC connection established in step 6. (The QUIC connection remains open until segment requests are received from the Media Player.)**

**8b: Pushed Initialization Segments are cached by the Media Player.**

15: **If not already pushed to and cached by the Media Player in step 8b,** the Media Player requests Initialization Segments from the 5GMSd AS. **Each segment request/response consumes a QUIC client-initiated stream in the HTTP/3 protocol mapping.**

17n: The Media Player sends requests for media segments **of the same group within the corresponding QUIC stream**. **If sent simultaneously, the Media Player may send priority information with or apply the prioritization to the media segment requests.**

18n: The 5GMSd AS responds to the requests of a same groups within the corresponding QUIC stream.

**19: Requests on the QUIC stream for each group operate in parallel in the same QUIC connection if they are targeting the same 5GMSd AS endpoint at reference point M4.**

END OF CHANGES