**3GPP TSG- Meeting #**

**, , -**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **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 |  | Radio Access Network |  | Core Network |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | New clause 5.X QUIC-based Media Delivery | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***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 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-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | FS\_AMD includes a new topic about opportunities with QUIC for segmented streaming (topic “m)”). After updating the clause on HTTP/3 (clause 5.4) of TR 26.804 with up-to-date information since some parts are common to the new topic, we introduce a new section with QUIC-specific media delivery aspects. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | The following changes are proposed:   * Add clause X.X QUIC-based Media Delivery | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  |  | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  |  | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  |  | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

Change #1

## 2 References

[2] Akamai Blog, "A QUICk Introduction to HTTP/3", April 2020, <https://www.akamai.com/blog/developers/a-quick-introduction-http3>

[5] IETF RFC 9114: "HTTP/3", June 2022

[W3C-WT] W3C Working Draft: "WebTransport", May 2024,  
[https://www.w3.org/TR/webtransport/#web-transport](https://www.w3.org/TR/webtransport/%23web-transport)

Change #2

5.X QUIC-based media delivery

5.X.1 Description

5.X.1.1 General

QUIC, as specified in RFC 9000 [32], is a secure, reliable, multiplexed, connection-oriented transport protocol built on top of UDP. It is widely available and its impact on media streaming requires further study. In this clause we study QUIC-specific media streaming, not necessarily based on HTTP/3 [5] that is studied in clause 5.4.

A QUIC client establishes a connection with a server, and within this connection can transport data through multiple streams. Thanks to a more efficient implementation of the TLS initial handshake, a QUIC connection is typically established faster than a TCP + TLS connection, therefore reducing initialization time. Additionally, by allowing logical streams to be multiplexed into a single QUIC connection, they can operate independently of each other, each with its own separate congestion window. Because QUIC is layered on top of UDP, a stall in one stream does not block progress in any others. Similarly, packet loss in one logical stream does not affect the progress of data transfer in other streams multiplexed in the same QUIC connection. Finally, QUIC, like HTTP/2 [4], supports prioritisation capabilities at a stream level.

Editor's Note: Further content to be provided.

Even though QUIC solves many issues compared to TCP (and HTTP/3 compared to HTTP/2 respectively), there are some open issues and shortcomings when it comes to media content delivery.

5.X.1.2 Application access to QUIC protocol features

Typically access to QUIC is mediated through a QUIC-enabled application protocol such as HTTP/3 [5] or WebTransport [W3C-WT]. For that reason, the set of QUIC features available to an application is the subset exposed by the chosen application protocol (typically invoked via the public API of a client or server library). This approach facilitates efficient application development and integration of new features but at the cost of limiting management of the connections and streams.

5.X.1.3 Connection and stream management

Most applications use QUIC because they are using HTTP/3 [5]. The mechanism for handling streams is tightly specified by the HTTP/3 protocol: each HTTP transaction consumes one logical stream in the QUIC connection.

Applications are encouraged to keep QUIC connections alive when it makes sense to do so, and HTTP/3 client libraries typically facilitate connection keep-alive behaviour for efficiency reasons. In addition, QUIC's "0-RTT" connection establishment procedure allows a client to reconnect to a server it has previously connected to, and to reuse a security context negotiated during a previous connection and cached by both parties to send application payload data to the server in the first UDP datagram of the new QUIC connection.

WebTransport [W3C-WT] allows more explicit management of connections and streams without however being studied yet in the context of segmented media delivery. WebTransport allows the application to request streams and directly write datagrams on the available streams. Finally, WebTransport supports different congestion control algorithms and unreliable connections.

Editor's Note: Further content to be provided.

5.X.1.4 Stream prioritisation

Even though QUIC supports prioritisation at a stream level, the way this prioritisation is applied is not standardised and it is left to implementation. Additionally, stream priority is nominated by the sender, and as such neither the network nor the recipient are aware of the applied mechanism.

#### 5.X.1.5 Key Issue objectives

5.X.1.5.1 Introduction

The key issues mentioned here are targeting QUIC-specific delivery aspects, those relevant to HTTP/3 have are addressed in Clause 5.4.

5.X.1.5.2 Application access to QUIC connections

Editor's Note: Further content to be provided.

5.X.1.5.3 Variability of QUIC implementations

Editor's Note: Further content to be provided.

Editor's Note: Focus the preceding set of open issues into a set of clear questions that this Key Issue aims to address.

5.X.2 Collaboration scenarios

Editor's Note: Further content to be provided - in TDoc S4-241571.

5.X.3 Architecture mapping

5.X.3.1 General

Editor's Note: Mapping against 4.1.2 Generalized Media Delivery architecture of TS26.501 [15] is to be provided.

5.X.4 High-level call flow

5.X.4.1 General

Editor's Note: Further content to be provided.

5.X.5 Gap analysis and requirements

5.X.6 Candidate Solutions

Editor’s Note: Provide candidate solutions (including call flows) for each of the identified issues.

5.X.7 Summary and Conclusions

The study of this Key Issue has explored the ways in which QUIC can be deployed to support the 5G Media Streaming architecture, and the potential open issues arising from this deployment.

Editor's Note: Further content to be provided.

END OF CHANGES