**3GPP TSG SA WG4 #113e *S4-210492r1***

**E-meeting, 6 - 14 April, 2021 Revision of S4-210298**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.0* | | | | | | | | |
| **Pseudo CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.8xx** | **CR** | **<CR#>** | **rev** | **1** | **Current version:** | **0.0.0** |  |
|  | | | | | | | | |
| *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\_5GMS-EXT] Key Topic Additional / New transport protocols | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Tencent | | | | | | | | | |
| ***Source to TSG:*** | SA4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_5GMS-EXT | | | | |  | ***Date:*** | | | 2021-03-30 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **B** |  | | | | | ***Release:*** | | | Rel-17 |
|  | *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-12 (Release 12)* *Rel-13 (Release 13) Rel-14 (Release 14) Rel-15 (Release 15) Rel-16 (Release 16)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Adding details for Section 5.4 to reflect HTTP/3 and QUIC potential open issues. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Update HTTP references and add QUIC-related references in Section 2.  Clarify relationship between "the main QUIC functions" and QUIC drafts that describe those functions in Section 5.4.1.  Add potential open questions in Section 5.4.1. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Key topic not addressed | | | | | | | | |
|  | |  | | | | | | | | |
| ***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:*** | |  | | | | | | | | |
| ***56*** | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

**===== CHANGE =====**

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] Akamai Blog, "A QUICk Introduction to HTTP/3", April 2020, <https://developer.akamai.com/blog/2020/04/14/quick-introduction-http3>

[3] Fielding, R., Nottingham, M., and J. Reschke, "HTTP/1.1", Work in Progress, Internet-Draft, draft-ietf-httpbis-messaging-15, 30 March 2021, http://www.ietf.org/internet-drafts/draft-ietf-httpbis-messaging-15.txt

[4] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, May 2015, https://www.rfc-editor.org/info/rfc7540

[5] Bishop, M. (Ed.), draft-ietf-quic-http-34, "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, 2 February 2021

[6] D. Bhat, A. Rizk, and M. Zink, "Not so QUIC: A Performance Study of DASH over QUIC," NOSSDAV'17: Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and VideoJune 2017 Pages 13–18 https://doi.org/10.1145/3083165.3083175

[7] AWS, "Achieving Great Video Quality Without Breaking the Bank", Streaming Media June 2019, [[https://pages.awscloud.com/rs/112-TZM-766/images/GEN elemental-wp-achieving-great-video-quality-without-breaking-the-bank.pdf](https://pages.awscloud.com/rs/112-TZM-766/images/GEN%20elemental-wp-achieving-great-video-quality-without-breaking-the-bank.pdf)](https://pages.awscloud.com/rs/112-TZM-766/images/GEN%20elemental-wp-achieving-great-video-quality-without-breaking-the-bank.pdf)

[8] Netflix, "Optimized shot-based encodes: Now Streaming!", Netflix Blog, May 2018, https://netflixtechblog.com/optimized-shot-based-encodes-now-streaming-4b9464204830

[9] DASH-IF/DVB Report on Low-Latency Live Service with DASH, July 2017, available here: <https://dash-industry-forum.github.io/docs/Report%20on%20Low%20Latency%20DASH.pdf>

[10] DASH-IF IOP Guidelines v5, Low-latency Modes for DASH, available here: <https://dash-industry-forum.github.io/docs/CR-Low-Latency-Live-r8.pdf>

[11] ISO/IEC 23009-1, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats"

[12] IETF RFC 8673, "HTTP Random Access and Live Content".

[13] 3GPP TR 26.939: "Guidelines on the Framework for Live Uplink Streaming (FLUS)".

[14] 3GPP TS 26.238: "Uplink Streaming".

[15] 3GPP TS 26.501

[16] 3GPP TS 26.512

[17] ISO/IEC 13818-1:2019 Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems

[18] SCTE STANDARD SCTE 35 2020 Digital Program Insertion Cueing Message <https://www.scte.org/pdf-redirect/?url=https://scte-cms-resource-storage.s3.amazonaws.com/SCTE-35-2020_notice-1609861286512.pdf>

[19] ISO/IEC 23000-19:2020 Information technology — Multimedia application format (MPEG-A) —Part 19: Common media application format (CMAF) for segmented media

[20] ISO/IEC 23009-1:2019/DAMD1 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats — Amendment 1: CMAF support, events processing model and other extensions [21] VSF TR-06-01, RIST Simple Profile, <https://www.videoservicesforum.org/download/technical_recommendations/VSF_TR-06-1_2018_10_17.pdf>

[22] VSF TR-06-02, RIST Main Profile, <https://www.videoservicesforum.org/download/technical_recommendations/VSF_TR-06-2_2020_03_24.pdf>

[23] 3GPP TS 23.501.

[24] 3GPP TS 23.502.

[25] 3GPP TS 29.517.

[26] C. Krasic, M. Bishop, and A. Frindell, Ed., draft-ietf-quic-qpack-21, "QPACK: Header Compression for HTTP/3", Work in Progress, Internet-Draft, 2 February 2021

[27] J. Iyengar, Ed. and M. Thomson, Ed., draft-ietf-quic-transport-34, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, 15 January 2021

[28] M. Thomson, Ed. and S. Turner, Ed., draft-ietf-quic-tls-34, "Using TLS to Secure QUIC", Work in Progress, Internet-Draft, 15 January 2021

[29] J. Iyengar, Ed. and I. Swett, Ed., draft-ietf-quic-recovery-34, "QUIC Loss Detection and Congestion Control", Work in Progress, Internet-Draft, 15 January 2021

[30] IETF RFC 5681, "TCP Congestion Control".

[31] M. Kuehlewind and B. Trammell, draft-ietf-quic-manageability-10, “Manageability of the QUIC Transport Protocol”, Work in Progress, Internet-Draft, 22 February 2021

[32] N. Cardwell et. al. “BBR Updates: Internal Deployment, Code, Draft Plans”, 9 March 2021, https://datatracker.ietf.org/meeting/110/materials/slides-110-iccrg-bbr-updates-00.pdf

**===== CHANGE =====**

## 5.4 Additional/new transport protocols

### 5.4.1 Description

Media streaming applications are continued to use HTTP-based distribution protocols, but newer versions of HTTP such as HTTP/2 or HTTP/3 are introduced, see for example also TR 26.925 [5], clause 6.1.4. The architectural and performance impacts of such protocols for 5G-based media distribution is unclear and requires study. The study also considers how Media Players may use functionalities existing in new transport protocols, and also investigate the impact of new transport protocols on 5GMS usage and traffic identification (e.g. Service Data Flow Descriptions).

Based on [X], HTTP protocol (also known as web protocol), powers most websites, mobile apps, and videos. It was created by Tim Berners-Lee at CERN in 1989 and has been enhanced over the years to keep up with the ever-changing World Wide Web. Currently, the web is a mixture of HTTP/1.1 [3] and HTTP/2 [4] adoption. Most well-known websites are running HTTP/2, while smaller websites and late adopters plan to migrate to HTTP/2 in the near future as it is relatively easy to implement. HTTP/2 is used by about 45% of websites and supported by all major web browsers. HTTP/3 is only used by about 5% of websites now and not well-supported by web browsers yet. However. significant HTTP/3 deployments are emerging. For example, YouTube™ has for a long time been offering a pre-RFC draft version to any client that wants to use it, especially the Chrome™ browser. Other browsers are expected to follow soon after waiting for the QUIC and HTTP/3 RFCs to be published before mainlining that feature.

HTTP/2 introduces the "Streams" concept at HTTP level and each stream can have different priorities. All objects can from a web-page can be multiplexed in single long-lived TCP connection. Also, HTTP/uses header compression (HPACK) to avoid verbose/clear text. Also, HTTP/2 pseudo-mandates TLS to prevent “middle boxes” from messing up with the content. However, HTTP/2 does not remove the drawbacks of TCP’s head-of-line blocking - packet loss on one stream will block all other streams until recovery even if packets for all other streams are correctly received.

HTTP/2 testing shows [2] that the delivery of large objects over HTTP/2 can be slower than over HTTP/1.1 when there is packet loss. This is because HTTP/2 uses a single TCP connection, versus about six connections which most web browsers open over HTTP/1.1. In addition, the TCP congestion control algorithms reduce the TCP congestion window size, resulting in fewer bytes sent over the wire when using just one TCP connection.

HTTP/2 provides on average a 5% to 15% performance improvement on page load times over HTTP/1.1. HTTP/1.1 allows persistent TCP connections, but requests still had to be serialized, resulting in the well-known "HTTP head of queue blocking". In order to improve downloads, many TCP flows still needed to be parallelized to speed up delivery.

The solution to this problem is to use HTTP/2 over a different transport protocol that provides more efficient congestion control. One option would be to upgrade and modify TCP, but modifying TCP implementations is viewed as an impossible task. For example, middle boxes such as NAT, Firewalls, and Load balancers are problematic, because they get rarely upgraded which prevents any updates to TCP. TCP is also hard to evolve as it is almost always implemented as part of operating system kernels, requiring an updated operating system as part of TCP updates. Hence, it was considered easier to introduce transport functions on top of UDP, outside the operating system kernel, in the user space – referred to as QUIC.

That, in essence, is what HTTP/3[5] is: HTTP/2 over User Datagram Protocol (UDP) based on IETF QUIC. HTTP/3 is a thin layer on top of QUIC [27] including QPACK header compression [26]. The main QUIC functions are connection and stream multiplexing [27], fast startup [27], in-order delivery (within stream)[27], flow control [27]. TLS1.3 (handshake) [28], loss recovery and congestion control [29]

By multiplexing multiple concurrent logical streams over a single UDP-based transport association, and by giving each stream its own independent byte offset numbering space, packet loss in one stream does not block progress on other logical streams in the same QUIC connection. (However, the affected stream will still block when packets are lost, so as to guarantee in-order delivery of payloads to the application.).

A screenshot of a cell phone

Description automatically generated

Figure 5.4-1: HTTP/2 and HTTP/3 protocol stacks

For an entertaining introduction to QUIC and HTTP/3, please check <https://www.youtube.com/watch?v=B1SQFjIXJtc>.

However, using QUIC for adaptive streaming still requires study as under certain circumstances, the quality using QUIC may even degrade for DASH-based streaming than it would increase [6]. The evaluation results show that using the unmodified DASH algorithms on top of QUIC may not provide the anticipated performance boost when compared to the standard DASH over TCP.

The main expected benefit of QUIC is being able to multiplex requests for all Adaptation Sets onto the same transport association, and then to manage the network QoS on that aggregate connection. This has a valuable operational benefit to a CDN operator (including the 5GMS AS) in reducing the number of UDP ports that a server needs to keep open. Another benefit is being able to migrate connections from one IP address to another with minimal interruption to either client or server. This is useful when the client moves, but it is also useful when the server changes (e.g. in edge computing relocation Use Cases).

Because HTTP/3 and IETF QUIC are new protocols, there are several questions that need to be investigated during this study.

The IETF specifications for HTTP/3 [5] and the core QUIC functions [26, 27, 28, 29] are now approved in the IETF, and have been broadly deployed by a number of browser vendors and content providers, since the IETF QUIC working group has focused on specification, implementation, and, after the specifications were sufficiently stable enough, deployment, all in parallel. The performance of HTTP/3 over IETF QUIC in environments that have not been encountered during deployments to date is still an open question. Of greatest interest for this study, is the performance of HTTP/3 over IETF QUIC in 5G networks. Although deployment of 5G networks has begun, most deployment experience with HTTP/3 over IETF QUIC in mobile networks over the past few years has been in non-5G networks.

One of the core functions of QUIC is the capability to migrate connections without application involvement when endpoint IP addresses change, rather than requiring the detection of a connection failure, teardown of that connection, and setup of a new connection. Connection migration is one of the key QUIC functions that we do not have a great deal of experience with – implementers in the IETF said they were concentrating on performance for a connection, and many had not completely implemented or tested connection migrations at scale (datapoint is from October 2020).

When end users have used HTTP/3 over QUIC to access servers outside the 5G core network, 3GPP terminal mobility was handled transparently by the 3GPP network, and the UE’s IP address(es) didn’t change. If a server’s IP addresses changed, this was often not visible to the user, due to the widespread deployment of CDNs and loadbalancers in data centers. If edge computing resource IP addresses change in relocation use cases, QUIC connection migration could be used to reduce the impact on user experience, but this needs to be analyzed carefully.

The standardized QUIC congestion control and recovery procedures in [29] are chosen to emulate TCP’s standardized behaviors ([30], plus extensions). These are quite conservative, and not match current work on delay-based congestion control and recovery mechanisms, which have also seen wide deployment in QUIC implementations. In principle, delay-based congestion control and recovery mechanisms should improve user experience for streaming media applications, but this isn’t known yet, and this needs to be carefully analyzed.

[6] raised the issue that MPEG-DASH performance might be lower over HTTP/3 than over HTTP/2. This reference was comparing Google’s pre-standardization QUIC implementation to highly optimized HTTP/2 over TCP implementations, and is about five years old, as of this writing, but the point remains – we need to know more about unmodified MPEG-DASH over standardized HTTP/3 implementations.

In particular, in the downlink media streaming Use Case, the server chooses the algorithm but the client media player decides whether to use a long-lived connection or to drop and reconnect when changing representation/rendition or adaptation/switching set, and the decision on connection (re)use is often delegated by the client application to a library, and this library might not offer an API to influence connection reuse. QUIC optimisations for reducing connection setup time (0-RTT or 1-RTT handshake (as described in [27]) mitigate this suboptimal use of connections to some extent, but the if the client, directly or indirectly by delegating the decision to the library, chooses to close and open connections, any QUIC congestion control algorithm begins probing for available bandwidth with no information about path characteristics. The connection handling characteristics of common APIs need to be analyzed as part of this study.

MPEG-DASH has provided years of good user experience running over HTTP/1.1. As use cases arise which require very low latency, it is reasonable to better understand how, and when, MPEG-DASH might take advantage of HTTP/3 and analyze whether this has any implications for this study.

One of the biggest distinctions between HTTP/2 over TCP and HTTP/3 over QUIC has been the encryption of almost all transport-level information in QUIC. This information, which was not encrypted in TCP even when it was carrying encrypted payloads, was often used in network management to identify and troubleshoot performance problems on the Internet. In most of experience with HTTP/3 over QUIC deployments to date, content providers (e.g. Google/YouTube, Facebook, etc.) have terminated at least one end of the encrypted end-to-end connections, allowing them to identify problems at the QUIC transport level. That might be true in 5G deployments, or it might not be the case. If that is not the case, it would be very useful to consider the guidance in [31] as part of this study. It is also likely that operator deployments which relied on split-TCP connections to improve performance over radio links will require reconsideration for QUIC-based t, since QUIC transport-level information is not available unless a device has a security context for the encrypted QUIC connection. It is likely that we need to enhance the 3GPP QoS framework, and that if an application is using QUIC, 5-tuples are not sufficient for per-flow QoS.

Although QUIC can be implemented as part of operating system kernels, it is commonly implemented in user space, allowing frequent updates to congestion control and recovery procedures, including introduction of entirely new procedures (e.g. BBR, "Bottleneck Bandwidth and Round-trip propagation time" congestion control [32]). It is likely that the performance characteristics of HTTP/3 applications will vary more dynamically than HTTP/1.1 implementations that have been used by MPEG-DASH in the past. This may also have implications for the existing

### 5.4.2 Collaboration Scenarios

A service provider/content provider runs an adaptive media streaming service between HTTP/3 and QUIC enabled 5G Media Streaming AS and an HTTP/3 and QUIC enabled UE using 5G Media Streaming over M2d and M4d.

Editor’s Note: Study collaboration scenarios between the 5G System and Application Provider for each of the key topics.

### 5.4.3 Deployment Architectures

Editor’s Note: Based on the 5GMS Architecture, develop one or more deployment architectures that address the key topics and the collaboration models.

### 5.4.4 Mapping to 5G Media Streaming and High-Level Call Flows

Editor’s Note: Map the key topics to basic functions and develop high-level call flows.

### 5.4.5 Potential open issues

### Editor’s Note: Identify the issues that need to be solved.5.4.6 Candidate Solutions

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