**3GPP TSG- Meeting # *r2***

**Online, 8th April 2024 - 12th April 2024**

|  |
| --- |
| *CR-Form-v12.2* |
| **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:***  | Updates on clause 5.4 Additional/new transport protocols |
|  |  |
| ***Source to WG:*** | , BBC |
| ***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 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-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)”). Before adding a new clause for this topic, it is necessary to update the current 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. Also the current tile 5.4 seemed vague and it should be refined to better reflect the content of the clause. |
|  |  |
| ***Summary of change:*** | The following changes are proposed:* Refining the title of clause 5.4 Additional/new transport protocols to Upgrading adaptive streaming to HTTP/3
* Updating information regarding standardisation status and usage of HTTP/3
 |
|  |  |
| ***Consequences if not approved:*** | No up-to-date information on HTTP/3, block the start of work for the new topic “m)”.  |
|  |  |
| ***Clauses affected:*** | 5.4 |
|  |  |
|  | **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

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

…

[x1] Belson, D. and Pardue L.: "Examining HTTP/3 usage one year on", June 2023
<https://blog.cloudflare.com/http3-usage-one-year-on>

[x2] IETF RFC 5321: “Simple Mail Transfer Protocol”, October 2008.

[x3] HTTP Archive: State of the Web, https://httparchive.org/reports/state-of-the-web#h3

CHANGE #2

## 5.4 Use of HTTP/3 in the Media Delivery System

CHANGE #3

### 5.4.1 Description

#### 5.4.1.1 General

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 [88], 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 SMTP (Simple Mail Transport Protocol) [x2] and reusing MIME (Multipurpose Internet Mail Extensions) notation [74][75], 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. In 2021, HTTP/2 was used by about 45% of websites and supported by all major web browsers while HTTP/3 was only used by about 5% of websites and was not then well-supported by web browsers. However, significant HTTP/3 deployments emerged which boosted the deployment of HTTP/3. For example, YouTube™ had for a long time been offering a pre-RFC draft version to any client that wanted to use it, especially the Chrome™ browser. Other browsers, including Mozilla Firefox, Microsoft Edge and Apple Safari™ followed soon after waiting for the QUIC [32] and HTTP/3 [5] RFCs to be published before mainlining that feature.

According to an analysis done by the CDN provider Cloudflare published in mid-2023 [x1], "mobile devices are responsible for over half of request volume to Cloudflare, with Chrome Mobile generating more than 25% of all requests, and Mobile Safari more than 10%". Table 5.4.1.1‑1 summarises the breakdown of traffic to Cloudflare endpoints per HTTP version for different web browser types during the month of March 2023.

Table 5.4.1.1‑1: Breakdown by browser type of HTTP version usage
against Cloudflare endpoints (March 2023)

|  |  |  |  |
| --- | --- | --- | --- |
| User Agent | HTTP/3 | HTTP/2 | HTTP/1.1 |
| Mobile Chrome | ~40% | ~35% | ~5% |
| Mobile Safari | ~20% | ~75% | ~5% |
| *All User Agents* | ~28% | ~63% | ~9% |
| NOTE: Data is for generic HTTP traffic andis not limited to streaming services. |

Editor’s note: Those data points are for generic HTTP traffic and similar data for streaming services would be desirable.

In the span of two years (2021–2023), the increase of HTTP/3 usage by web browser User Agents has been significant.According to a report published by the HTTP Archive [x3], the percentage of websites supporting HTTP/3 has also been increasing during the same period. Table 5.4.1.1-2 presents the percentage of web sites supporting HTTP/3 for December 2021 and March 2024 using Chrome on mobile and desktop platforms to collect the data.

Table 5.4.1.1‑2: Percentage of web sites supporting HTTP/3
as surveyed by the HTTPArchive

|  |  |
| --- | --- |
| Web Browser | Percentage of websitessupporting HTTP/3 |
|  | December 2021 | March 2024 |
| Mobile Chrome | 12.6% | 28.4% |
| Desktop Chrome | 12.2% | 27.0% |

Editor’s note: Those data points are for generic web sites and similar data for streaming services would be desirable.

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 [2]. 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 a new transport protocol on top of UDP, that can be implemented outside the operating system kernel, in the user space. This new transport protocol is 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 [32] including QPACK header compression [31]. The main QUIC functions are connection and stream multiplexing [32], fast startup[32], loss recovery, in-order delivery (within stream) [32], flow control [32]. TLS1.3 (handshake) [33], loss recovery and congestion control [34].

HTTP/3 always uses the QUIC protocol as its transport layer, although QUIC may also be used to carry other application-level protocols. For 5MBS, the term “HTTP/3” will always be used to refer to “HTTP/3 over QUIC”, unless the text refers specifically to QUIC in explaining its effect on HTTP/3.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.).



Figure 5.4.1.1-1: 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 HTTP/3 over 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 about performance and management that need to be investigated during this study.

END OF CHANGES