network

Article

Implementation and Evaluation of HTTP/3 Connectivity Check Using Happy Eyeballs Algorithm

Tomofumi Matsuzawa * and Kyosuke Ichikawa

Department of Information Sciences, Faculty of Science and Technology, Tokyo University of Science, Yamazaki, Chiba 278-8510, Japan; papers@mail.ebi-yade.com

* Correspondence: t-matsu@is.noda.tus.ac.jp

Abstract: The HTTP Alternative Services (Alt-Svc) method is defined as an application to check connectivity in HTTP/3. This method is designed based on the fact that communication with old HTTP is guaranteed and the HTTP/3 adoption rate is not necessarily dominant, and it is considered effective in the early stages of transition. However, once HTTP/3 has reached its peak and the transitional period has passed, the uncertainty and redundancy of the Alt-Svc procedure become detrimental. In Alt-Svc, the procedure involves first completing the old HTTP connection to use HTTP/3, and then migrating to HTTP/3 if possible; however, because HTTP/3 is a protocol that eliminates the waste of the old HTTP handshake (TCP handshake followed by TLS handshake), HTTP/3 does not fully benefit from the rapid connection establishment of HTTP/3. Therefore, we propose a method to apply the Happy Eyeballs algorithm, which is used for IPv4 and IPv6 connectivity checks, to the old HTTP and HTTP/3 connectivity checks. The Happy Eyeballs algorithm performs the two selections in parallel to eliminate the delay that occurs in sequential processing, but the proposed method differs from the conventional Happy Eyeballs algorithm in that, even if the old HTTP is adopted once, it switches to the HTTP/3 connection if it is possible to connect using HTTP/3. The proposed method differs from the conventional Happy Eyeballs algorithm by introducing a mechanism to switch to HTTP/3 connections when HTTP/3 connections are available, even when the old HTTP is adopted. Results of the evaluation experiments demonstrated that the adoption rate of HTTP/3 increases in environments with high communication latency because the old HTTP performs the TLS handshake after the TCP handshake, but with this improvement, HTTP/3 is preferentially selected even in low latency environments when it is selectable.

Keywords: QUIC; HTTP/3; connectivity check; Happy Eyeballs

check for updates

Citation: Matsuzawa, T.; Ichikawa, K. Implementation and Evaluation of HTTP/3 Connectivity Check Using Happy Eyeballs Algorithm. *Network* **2022**, 2, 389–397. https://doi.org/10.3390/network2030024

Academic Editor: Andreas Kassler

Received: 10 May 2022
Accepted: 24 June 2022
Published: 28 June 2022

1. Introduction

On 31 May 2021, the QUIC protocol was standardized as RFC 9000 [1] by the Internet Engineering Task Force (IETF). It has attracted interest along with HTTP/3, which is a new application protocol based on QUIC. According to w3tech usage statistics [2], approximately 25% of web sites use HTTP/3 as an element, and over 75% of Internet traffic on Meta's own services is HTTP/3 and QUIC. Thus, QUIC has received support from vendors with large customer bases on the real Internet and is expected to expand further in the future. For HTTP/3 connectivity checks, RFC9113 [3] already stipulates the use of the HTTP Alternative Services (Alt-Svc) [4] method. This method is designed within the context of guaranteed communication over old HTTP and the fact that HTTP/3 penetration is not necessarily dominant; it is considered as effective in the early stages of transition. However, when HTTP/3 is well established and the transitional period has passed, the uncertainty and redundancy of the Alt-Svc procedure are detrimental. In Alt-Svc, to use HTTP/3, it is necessary to first complete the old HTTP connection, and then migrate to HTTP/3 if possible. As HTTP/3 is a protocol that eliminates the waste of the old HTTP handshake (TCP handshake followed by TLS handshake), the current Alt-Svc does not fully benefit from the rapid connection establishment of HTTP/3. Alt-Svc, which connects using the old HTTP and then uses HTTP/3 if possible, is a suitable method during the transition period when the old HTTP is the mainstream. However, even if HTTP/3 is used extensively in the future, web servers must always respond to requests for the old HTTP because the connection must first be completed using the old HTTP. Moreover, because the HTTP/3 connection is initiated once the old HTTP connection has been completed, Alt-Svc is an inefficient method. Therefore, a method that attempts HTTP/3 connections from the beginning is necessary for the latter part of the transition period to HTTP/3. In this study, we propose a connectivity check for the latter half of the HTTP/3 transition period, with the aim of aiding the transition to HTTP/3.

2. Related Works

2.1. QUIC and HTTP/3

QUIC [1] is a transport layer protocol based on the User Datagram Protocol (UDP) [5]. In contrast to the Transmission Control Protocol (TCP) [6], the UDP includes relatively few functions, such as packet reordering, error detection, retransmission, congestion control, and handshaking. QUIC takes advantage of this, absorbs the above TCP features, and reimplements them in a single protocol that is integrated with Transport Layer Security (TLS) [7]. Moreover, QUIC implements multiplexing of connections that are newly created by HTTP/2 [8], and it reduces the responsibilities of the application layer in achieving thorough efficiency. HTTP/3 is a new major version of HTTP, the specification which is developed on the premise of QUIC, and it is simpler than the old HTTP. The basic syntax is compatible, and the fundamental difference between the improved and old HTTP lies in the use of the TCP or QUIC for the transport layer protocol. As the transport layer protocols differ, the incompatibility with the network environment is significant, and the connectivity problem, which is the subject of our study, is increasingly important. An overview of the new and old HTTP protocols is depicted in Figure 1.

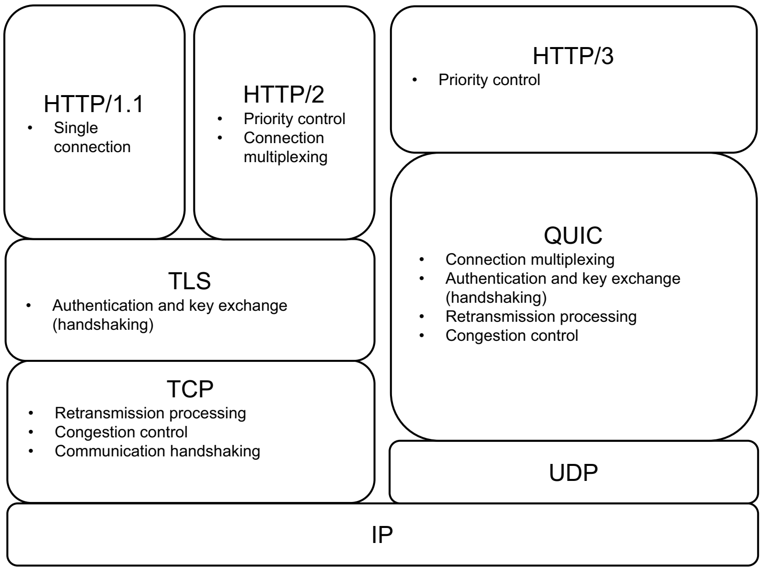
HTTP/1.1
• Single connection
HTTP/2
• Priority control
• Connection multiplexing
HTTP/3
• Priority control
TLS
• Authentication and key exchange (handshaking)
QUIC
• Connection multiplexing
• Authentication and key exchange (handshaking)
• Retransmission processing
• Congestion control
TCP
• Retransmission processing
• Congestion control
• Communication handshaking
UDP
IP

Figure 1. Overview of new and old HTTP protocols.

2.2. Current Connectivity Verification Method: Alt-Svc

The current method, namely Alt-Svc [4], was originally developed to perform the transition from HTTP/1.1 to HTTP/2. The specific connection confirmation procedure is as follows:

1. The client makes an HTTP request using the old protocol.
2. The server sets the corresponding protocol version, address, etc. in the "Alt-Svc" field (The "Alt-Svc" field may contain cache information, in which case the client

uses it based on the given expiration date, unless it detects a change in the network configuration) of the response header and sends the request using the old protocol.

3. The client sends the HTTP request using the new protocol from the next request based on the information obtained from the "Alt-Svc" field.

Furthermore, in the context of HTTP/3, the specification stipulates that clients fall back on the old HTTP if they cannot connect to HTTP/3 [3,9]. Three primary problems with the Alt-Svc procedure exist. First, the fact that the response header that is sent with the old protocol contains connection information for the new protocol guarantees software support of the server itself; however, this does not guarantee that the intervening network path accepts the new protocol. That is, if the server supports the new protocol but the network path does not, the communication after the protocol switchover waits for a timeout. This phenomenon cannot be ignored in the context of HTTP/3, which is incompatible with the transport layer. Second, whereas QUIC eliminates duplicate handshakes to save time in initiating communication, this procedure requires communication in the old protocol first (unless there is a valid cache), which indicates that this benefit is not available. Finally, both the server and client are required to support the old protocol. If a new server is built using only the new protocol, it will be inaccessible to clients using Alt-Svc (even if the client implements the new protocol). Given the above, the first of these problems is likely to become more pronounced at the onset of protocol transition, and the second and third problems are likely to become enhanced after the transitional period. Hence, Alt-Svc is not a general-purpose method that can be effective in the long term.

2.3. Happy Eyeballs

Happy Eyeballs [10] is an algorithm that was developed to deal with the fallback problem (Latency and connectivity issues when switching to IPv4 after the IPv6 connection fails) that is caused by the mixing of IPv4 and IPv6 environments. As it assumes that TCP is selected as the transport layer protocol, it cannot be directly applied to HTTP/3, which uses UDP-based QUIC. However, there exist similarities with a previous study regarding the objective of addressing connectivity issues in environments where incompatible protocols coexist. This algorithm is also actively used by major browsers and is a common method for IPv4/IPv6 fallback. Without Happy Eyeballs, it is necessary to attempt a TCP connection on one side (e.g., IPv6) first, and then on the other if it fails completely. However, this method tends to result in a waiting time of tens of seconds when the first attempt fails. The greatest advantage of adopting Happy Eyeballs is that it prevents the foregoing and improves the end-user experience. Conversely, during the several steps to be followed before the TCP handshake is established, IPv4 and IPv6 communicate twice, which certainly increases the overall network traffic volume. This is burdensome for service providers and backbone communication networks, and is thus cited as a disadvantage. The complexity of the software to be implemented is another qualitative disadvantage owing to the occurrence of parallel processing. An overview of the process is illustrated in Figure 2.

First, the DNS is queried for A and AAAA records. As these are in IPv4 and IPv6 record format, if both addresses are obtained, it is known that this is a dual-stack environment where both addresses are available. In this case, asynchronous TCP connection attempts are made over IPv4 and IPv6, and the one that completes the TCP handshake earlier is adopted. The connection attempts should not start at exactly the same time, and a delay of approximately 150–250 ms is recommended. if the delay is too large, it will not improve the user experience that HappyEyeballs attempts to provide. If too small (on the order of tens of milliseconds), it will achieve this goal, but at the cost of network traffic. Therefore, the RFC6555 document states, "It is RECOMMENDED that connection attempts be paced 150–250 ms apart to balance human factors against network load." The name Happy Eyeballs is derived from the improved end-user experience.

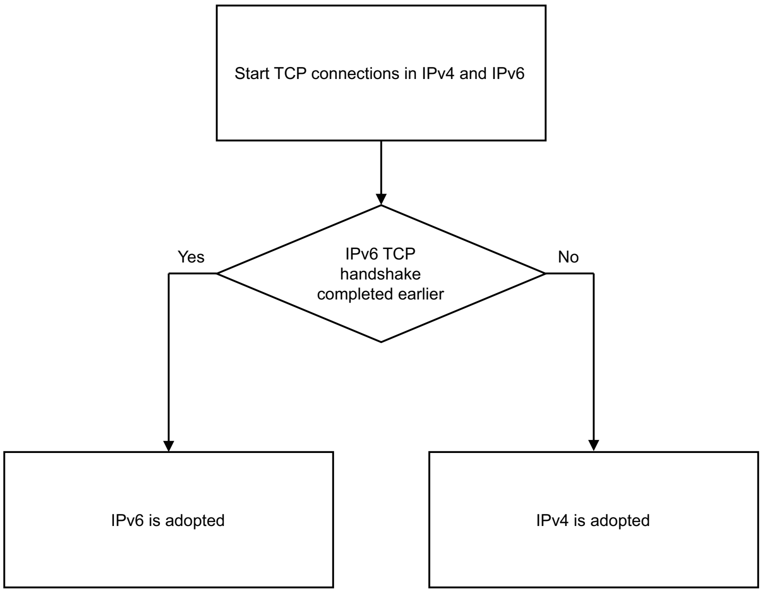
Start TCP connections in IPv4 and IPv6
IPv6 TCP handshake completed earlier
Yes
No
IPv6 is adopted
IPv4 is adopted

Figure 2. Happy Eyeballs.

3. Proposed Method

We propose a method for the application of the Happy Eyeballs algorithm to HTTP/3. However, as mentioned in Section 2.3, it cannot be applied immediately and several changes are required. Furthermore, this method offers several medium- and long-term benefits that will arise later, in addition to the short-term speedup until communication starts, owing to features such as QUIC congestion control, retransmission control, and HTTP priority control. We assume that QUIC encrypts the communications with the TLS so that the `http` schema indicating unencrypted communications is not used, and URLs are presented in the `https` schema. Moreover, we assume that the HTTP/3 acceptance port number is fixed to 443. Subsequently, the connection start delay discussed in Section 5 of the Happy Eyeballs specification [11]; i.e., the time between starting HTTP/3 and the old HTTP in this method, is fixed to 10 ms, which is defined as the minimum line. Other recommendations include a range of 100 ms–2 s depending on the congestion, with a default of 250 ms; however, this study is an application to a problem other than IPv4/IPv6. This means that the recommended value cannot be used as is. Moreover, there is no scientific basis for the original value, and it only indicates that the one round trip time (RTT) is a guideline. The IPv6 value is also not used as is. Furthermore, although the new protocol is given priority in this study, the old protocol should be similarly explored with little delay because the share of QUIC-enabled sites on the Internet is not high. After considering the intent of the specification to the extent applicable to this study, a fixed delay of 10 ms was selected to avoid computational problems. Moreover, the 0-RTT (this method sends application data from the beginning, skipping the TLS or other handshake for the communication destination that is obtained from the previous connection information. Originally proposed in RFC8446 [7], which defines TLS1.3, it was also incorporated into QUIC as a specification) function of QUIC is not used. This is because it assumes that the QUIC connection is available, which is beyond the scope of this study, and that the control for secure communication is complicated.

In the Happy Eyeballs evaluation experiment on IPv4/IPv6 that was conducted by Bajpai et al. [12], the authors compared the timings of the end of the TCP handshakes. However, in this context, the new protocol and old protocol have different transport layers, so the timing must be considered to compare the connectivity. This method compares the point in time when the QUIC-aggregated handshake ends in the new protocol with the point in time when both the TCP and TLS overlapping handshakes end in the old protocol. There are several reasons for this: the above are aligned in timing, application data are sent and received, and end users strongly prefer a final conclusion and not an intermediate result of the connection. The specific logic is presented in Figure 3.

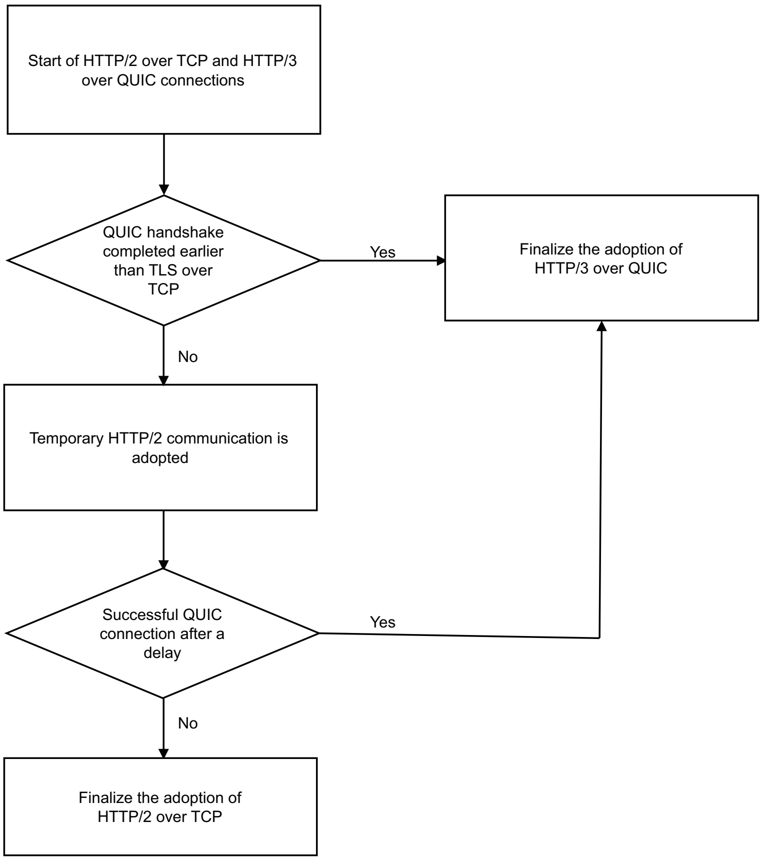
Start of HTTP/2 over TCP and HTTP/3 over QUIC connections
QUIC handshake completed earlier than TLS over TCP
Yes
Finalize the adoption of HTTP/3 over QUIC
No
Temporary HTTP/2 communication is adopted
Successful QUIC connection after a delay
Yes
No
Finalize the adoption of HTTP/2 over TCP

Figure 3. Flowchart of proposed method.

After a 10 ms connection start delay, HTTP/3 and old HTTP connection attempts are started in succession. If the QUIC handshake ends first, HTTP/3 is adopted and the other process is terminated immediately. Otherwise, the old HTTP is adopted and the HTTP request proceeds. If QUIC fails to connect following a delay, the site in question is assumed to be unsupported by HTTP/3 and the information is stored in the client. If QUIC completes the connection after a delay, the first HTTP request scheduled on QUIC is cancelled to eliminate duplication, but the fact that the connection is possible via HTTP/3 is saved to the client. This allows all requests for a period during which a connection does not need to be reestablished to select the appropriate protocol by simply referring to the correspondence information that is stored on the client.

As a result of the above, all requests for a period that do not require reconnection can select the appropriate protocol simply by referring to the correspondence information that is stored in the client. The best approach is to store the client's correspondence information in the memory by creating a dictionary, wherein the key is a `host`, as described in RFC 3986 [13].

Unlike the conventional Happy Eyeballs, the proposed method uses HTTP/3 as the preferred connection algorithm. As indicated in Figure 3, even once the old HTTP is adopted, HTTP/3 is adopted if the connection by QUIC is completed later, because HTTP/3 has the advantage of being adopted preferentially. Furthermore, as with the conventional Happy Eyeballs, double communication occurs until the connection is established, so there is the problem of increased network traffic volume compared to the connection method that does not also use the proposed method.

4. Implementation and Experimental Results

4.1. Implementation

The logic of the proposed method was implemented in the client program (https://github.com/ebi-yade/quic-go, accessed on 10 May 2022) by extending the existing HTTP/3 package in the Go language. Furthermore, because we could not find any programs that supported the client side of the existing Alt-Svc method, at least in the Go language, we implemented a parser (https://github.com/ebi-yade/altsvc-go, accessed on 10 May 2022) that roughly met the specification. In this experiment, we evaluated the completion time of the initial handshake, and not the final HTTP/3 arrival time. Therefore, the program itself was embedded in the implemented client and its operation was confirmed, but the computation time and other factors resulting from the parsing did not affect the experimental results. The server program was built by applying a patch of the HTTP/3 implementation in Rust by Cloudflare (https://github.com/cloudflare/quiche, accessed on 10 May 2022) to NGINX5 version 1.16.1.

4.2. Experimental Methods

The execution environment was built with two Docker containers on the same network so that failure factors such as the external delay and packet loss could be intentionally reproduced in the outbound traffic of each container using the tc command. However, whereas the intended delay could be reproduced by imposing half of the burden on both for the external delay, the packet loss was a probability, and setting the same amount of load on both resulted in a squared survival rate. Therefore, in this experiment, when reproducing the packet loss, only the intended value was set, as it was on the server side and not on the client side. For the environment that was used as the base for this experiment, we formulated a construction method and disclosed it to the public (https://github.com/ebi-yade/network-compose, accessed on 10 May 2022). The specific experimental procedure is detailed below.

1. We measured the HTTP/3 adoption rate and connection establishment time using the proposed method in an environment without intentional failures.
2. External delay and packet loss were set separately, and the changes in these metrics were observed.
3. As a comparison experiment, we measured the connection establishment time in Alt-Svc. However, for a fair and clear evaluation, only the first request was repeatedly measured from the state in which no connection was established for both cases. (We did not measure whether HTTP/3 was adopted in the second and subsequent communications.)

First, the HTTP/3 adoption rate and connection establishment time using the proposed method were measured in an environment where no intentional failures existed. Subsequently, external delay and packet loss were set separately, and changes in these indicators were observed. Thereafter, as a control experiment, we measured the connection establishment time in Alt-Svc. However, to ensure a fair and unambiguous evaluation, only the first request was measured repeatedly from the state in which no connection was established for both cases. That is, we did not determine whether or not HTTP/3 was adopted in the second and subsequent communications. In principle, 1000 consecutive trials under the same conditions (represented as a single column in the table below) were performed synchronously.

4.3. Results of External Delay Setting

The HTTP/3 adoption rate and connectivity establishment time when external delays were intentionally generated are presented in Table 1, and the measured connectivity establishment time for Alt-Svc is displayed in Table 2. Herein, σ represents the standard deviation.

Table 1. HTTP/3 adoption rate and connection establishment time with external delay settings. (Happy Eyeballs).

Delay (ms)	Adoption Rate (%)	Avg. Time (ms)	σ (ms)
0	99.7	3.2	1.4
50	100.0	57.4	4.4
100	100.0	108.2	2.7
150	100.0	158.4	2.8
200	100.0	208.4	4.6

Table 2. HTTP/3 adoption rate and connection establishment time with external delay settings. (Alt-Svc).

Delay (ms)	Avg. Time (ms)	σ (ms)
0	2.1	0.8
50	109.9	4.9
100	209.6	3.9
150	310.4	3.2
200	410.5	7.4

4.4. Results of Packet Loss Setting

The HTTP/3 adoption rate and connection establishment time when the packet loss was intentionally reproduced are displayed in Table 3. The maximum values are also presented because the standard deviations were large.

Table 4 presents the measured connectivity establishment time for Alt-Svc, which was performed as a control experiment. Note that when the loss ratio was set to 50% or higher, Alt-Svc failed owing to a timeout (30,000 ms), and it was difficult to establish 1000 consecutive connections, which is why the results are not shown.

Table 3. HTTP/3 adoption rate and connection establishment time with packet loss settings. (Happy Eyeballs).

Loss Rate (%)	Adoption Rate (%)	Avg. Time (ms)	σ (ms)	Max. Time (ms)
0	99.7	3.2	1.4	20
3	96.4	4.8	32.4	1005
10	91.0	18.5	110.2	1079
25	78.0	79.3	278.0	3115
50	62.7	474.0	975.0	15,475

Table 4. HTTP/3 adoption rate and connection establishment time with packet loss settings. (Alt-Svc).

Loss Rate (%)	Avg. Time (ms)	σ (ms)	Max. Time (ms)
0	2.1	0.8	18
3	36.8	173.1	1073
10	120.6	310.2	3079
25	374.0	675.9	7125

Table 4 shows the measured connectivity establishment time for Alt-Svc, which was performed as a control experiment. Note that when the loss ratio was set to 50% or higher, Alt-Svc failed owing to a timeout (30,000 ms), and it was difficult to establish 1000 consecutive connections, which is why the results are not shown.

5. Discussion

In the environment without delays or losses, the time that was required to establish a connection was slightly longer for HTTP/3, but HTTP/3 had a higher probability of being adopted. The reason for the slight reversal in the connection establishment time may be that the network distance was extremely close, and the difference in processing time by the computers was noticeable. Because the proposed method processes multiple communications asynchronously, the computational burden is somewhat greater than that in existing methods. However, situations in which such delays and losses do not exist rarely occur. Even if they do occur, selecting HTTP/3 and ignoring the average 1 ms difference in the connection initiation time cannot pose a problem. It can be observed from Table 1 that even if external delays were to occur, selecting the Happy Eyeballs protocol would have little impact. In contrast, when external delays of 50 ms or more occured, the difference in the connection establishment time between the proposed and existing methods was close to the external delay time that was imposed on a single round-trip communication. This is the difference that is created by the handshake duplicated by the old HTTP in the existing method. That is, this similarity can be interpreted as a confirmation of the theory of efficiency to which the proposed method appeals. In particular, of the two failure factors that were targeted in this study, the external delay mimics the distance in a normal real network. Hence, it is quite possible that traffic on the real Internet may follow the experimental results as a function of the distance. We now consider the behavior in environments in which packet loss occured, which was mentioned as another failure factor. It appears from Table 3 that the adoption rate of HTTP/3 by Happy Eyeballs monotonically decreased as the packet loss rate increased. However, theoretically, this trend is reasonable because HTTP/3 is highly effective in packet retransmissions once the connection is established. Moreover, because the existing method failed to measure availability below 99.9% when the packet loss rate reaches 50%, the proposed method, which succeeded in 1,000 connections out of 1,000 under the same conditions, could be sufficiently valuable. Nevertheless, there is no doubt that HTTP/3 and QUIC are necessary protocols in situations where packets have a high probability of being lost and frequent retransmissions are required. Therefore, the development of a method in which HTTP/3 is selected more preferentially according to information such as the packet loss rate stored with the client remains a challenge for future work. Active discussions are ongoing on HTTP/3 connectivity checks, such as Iyengar's talk [14] on the Happy Eyeballs algorithm for using QUIC on Facebook, and a method to check the existence of HTTP/3 in the DNS Resource Record [15]. It is necessary to continue to provide various options for site administrators and users to select the most appropriate option for their environment.

6. Conclusions

In the early period of transition to HTTP/3, when the old HTTP is the mainstream, the current Alt-Svc method is considered as sufficient. However, when HTTP/3 begins to become the mainstream, Alt-Svc, which assumes connections to the old HTTP, is inefficient. Therefore, we have proposed a connectivity verification method for use during the transition period when HTTP/3 will become the mainstream. We improved the Happy Eyeballs algorithm and introduced a mechanism to use HTTP/3 preferentially in environments where HTTP/3 is available. In the evaluation experiment, we conducted connectivity tests using the proposed method on web servers that support both old HTTP and HTTP/3, and confirmed that HTTP/3 is selected with a high probability in environments where HTTP/3 is available, even when the old HTTP is used by the Happy Eyeballs algorithm. Unlike Alt-Svc, the proposed method can connect to web servers that do not support the old HTTP, and we believe that it will be an effective method in the latter half of the period of transition to HTTP/3. In the middle of the transition period, it is necessary to consider the combination of Alt-Svc and the proposed method, as well as the selection of the connectivity check method; thus we will consider this as future work. We will also investigate the application of the proposed method to methods other than HTTP.

Author Contributions: Conceptualization, T.M. and K.I.; methodology, T.M. and K.I.; software, K.I.; validation, T.M. and K.I.; formal analysis, T.M. and K.I.; investigation, T.M. and K.I.; data curation, K.I.; writing—original draft preparation, T.M.; writing—review and editing, T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy issues. It is also possible to reproduce the study using the software available at the GitHub link introduced in the footnote of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Iyengar, J.; Thomson, M. *QUIC: A UDP-Based Multiplexed and Secure Transport*; IETF RFC 9000; Internet Engineering Task Force: Washington, DC, USA, 2021.
2. Usage Statistics of HTTP/3 for Websites. Available online: https://w3techs.com/technologies/details/ce-http3 (accessed on 10 May 2022).
3. Thomson, M.; Benfield, C. *Hypertext Transfer Protocol Version 3 (HTTP/3)*; IETF RFC 9113; Internet Engineering Task Force: Washington, DC, USA, 2022.
4. Nottingham, M.; McManus, P.; Reschke, J. *HTTP Alternative Services*; IETF RFC 7838; Internet Engineering Task Force: Washington, DC, USA, 2016.
5. Postel, J. *User Datagram Protocol*; IETF RFC 768; Internet Engineering Task Force: Washington, DC, USA, 1980.
6. Postel, J. *Transmission Control Protocol*; IETF RFC 793; Internet Engineering Task Force: Washington, DC, USA, 1981.
7. Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3*; IETF RFC 8446; Internet Engineering Task Force: Washington, DC, USA, 2018.
8. Belshe, M.; Peon, R.; Thomson, M. *Hypertext Transfer Protocol Version 2 (HTTP/2)*; IETF RFC 7540; Internet Engineering Task Force: Washington, DC, USA, 2015.
9. Langley, A.; Riddoch, A.; Wilk, A.; Vicente, A.; Krasic, C.; Zhang, D.; Yang, F.; Kouranov, F.; Swett, I.; Iyengar, J.; et al. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication (ACM SIGCOMM '17), Los Angeles, CA, USA, 21–25 August 2017; pp. 183–196. [CrossRef]
10. Wing, D.; Yourtchenko, A. *Happy Eyeballs: Success with Dual-Stack Hosts*; IETF RFC 6555; Internet Engineering Task Force: Washington, DC, USA, 2012.
11. Schinazi, D.; Pauly, T. *Happy Eyeballs Version 2: Better Connectivity Using Concurrency*; IETF RFC 8305; Internet Engineering Task Force: Washington, DC, USA, 2017.
12. Bajpai, V.; Schönwälder, J. Measuring the Effects of Happy Eyeballs. In Proceedings of the 2016 Applied Networking Research Workshop, Berlin, Germany, 16 July 2016; pp. 38–41. [CrossRef]
13. Berners-Lee, T.; Fielding, R.; Masinter, L. *Uniform Resource Identifier (URI): Generic Syntax*; IETF RFC 3986; Internet Engineering Task Force: Washington, DC, USA, 2005.
14. Iyengar, S. Moving fast at scale: Experience deploying IETF QUIC at Facebook. In Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC (EPIQ'18), Heraklion, Greece, 4 December 2018; ACM: New York , NY, USA, 2018. [CrossRef]
15. Schwartz, B.; Bishop, M.; Nygren, E. *Service Binding and Parameter Specification via the DNS (DNS SVCB and HTTPS RRs)*; IETF draft-ietf-dnsop-svcb-https-09; Internet Engineering Task Force: Washington, DC, USA, 2022.