

Happy Optometrist: A Summary of the Development and Implementation of Happy Eyeballs

Tim Albrecht, Tim Betzer*

**Chair of Network Architectures and Services*

School of Computation, Information and Technology, Technical University of Munich, Germany

Email: tim.albrecht@mytum.de, betzer@net.in.tum.de

Abstract—

IPv4 address space has become increasingly exhausted and workarounds such as Carrier Grade NAT have had to be introduced. This has led to IPv6 proportionally gaining importance. While IPv6 offers a larger address space, built-in security options, more efficient routing, as well as many other advantages, its adoption has been slow and riddled with problems. As a way to ease this transition, many clients and services have opted for a dual-stack approach, which aims to enable connectivity via both protocols.

Happy Eyeballs (HE) was introduced as a way to prioritize IPv6 in these setups while not worsening user experience. This is achieved by keeping IPv4 as a viable backup option in case of IPv6 failure. The first version of HE was proposed as a standard by the IETF in 2012. Since then HEv2 has replaced HEv1 and a draft of a third version is currently under development.

This paper aims to summarize the development of the HE protocol and related works, as well as evaluate the current state of HE and its function in the modern internet landscape. The main focus will be on the parameters defined in HE, its relation to newly introduced protocols like QUIC and how the basic concept of HE could be adopted in more contexts.

Index Terms—Happy Eyeballs, IPv6, Dual-Stack

1. Introduction

Dual-stack setups have established themselves as one of the most important tools in supporting the transition from IPv4 to an IPv6-only internet. By allowing connections with both address families, clients can use IPv4-only, IPv6-only and other dual-stack services while providers can offer their services to clients who might only support one of the two address families. One might assume that this practice would result in an overall improved user experience, since, if both client and service use a dual-stack setup, two options for establishing a connection are offered.

1.1. Dual-stack before Happy Eyeballs

The state of dual-stack devices before the introduction of HE is explained in detail by Bajpai and Schönwälder in [2], [3]. They find that in 2009, only around ~1% of ALEXA 1M websites announced AAAA entries. Of these, about 40% were not accessible via IPv6. Furthermore,

for 12% of all websites with AAAA entries, more than half of their content failed over IPv6. References [2], [3] attribute this brokenness mostly to the majority of IPv6 connections using Teredo [4] or 6to4 [5], as shown by Google [1]. Native traffic overtook these in about March 2010 (see Fig. 1a). Additional studies [6] by Colitti et al. and [7] by Zander et al. find that “even in situations where relays work, Teredo / 6to4 add noticeable latency when compared to native IPv4 and IPv6” [3]. Of course, some innate brokenness can also be attributed to native IPv6 [8].

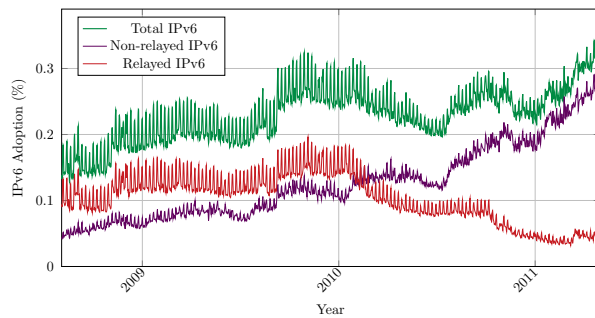
Still, the majority of devices tried establishing a connection over IPv6 first. This behavior is explained by the usage of `getaddrinfo()` [9], which returns a list containing both A and AAAA records. These are then ordered according to [10], resulting in all IPv6 addresses being pushed to top of the list. Consequently, any service aiming to establish a connection would try all IPv6 addresses first before falling back to IPv4.

This convention affected user experience greatly. Repeated connection attempts over broken IPv6 were causing delays of around 20 s to 40 s, as was analyzed by Chown and Morse in [11], while a connection over IPv4 would have succeeded as normal. This behavior is extremely harmful, since it would discourage clients and hosts alike from using a dual-stack setup, thereby sabotaging the transition from IPv4 to IPv6.

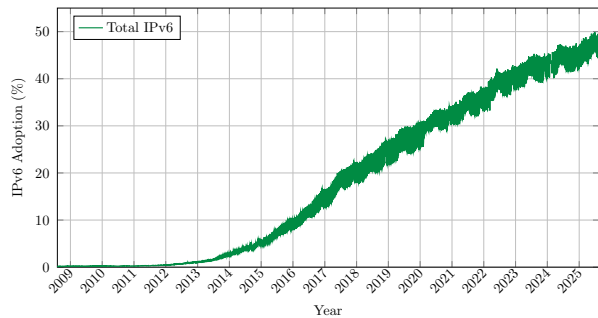
Corroborating these findings, Google’s IPv6 adoption statistics [1] show that less than 1% of all connections reaching Google were made over IPv6 until about the first quarter of 2013 (see Fig. 1).

1.2. Happy Eyeballs

The first version of the Happy Eyeballs protocol [12] was published by the IETF in 2012. As an RFC it proposes an internet standard that aims to improve user experience when working with dual-stack setups. The name stems from the goal to “make users’ eyeballs happy” by reducing waiting time for websites to load. HEv1 defines that if a connection attempt over one address family does not succeed, another using the remaining one should be started quickly after. A recommended value for this delay will be discussed in 2.1. Since IPv6 is preferred, it will be the address family that is tried first. In case of failure, a fallback to IPv4 would only cause a few milliseconds of delay, massively improving user experience compared to the 20 s to 40 s of delay mentioned in [11].



(a) From August 2008 to May 2011



(b) From August 2008 to August 2025

Figure 1: Percentage of connections reaching Google over IPv6. Figure reproduced from [1]

The next iteration, HEv2 [13], was proposed in 2017. Concepts, such as the delay between connections, are further specified and additional parameters such as a Resolution Delay and a First Address Family Count were added. These help in further prioritizing IPv6 while still ensuring a safe and fast fallback to IPv4. Sattler et al. [14] show that HEv2 is not yet widely implemented.

HEv3 [15] is the latest version, being worked on by the IETF since 2023. It focuses less on the specific parameters of HE implementations and more on the selection of additional DNS records, which are SVCB and HTTPS. This influences the L4 protocol being used and the ordering of the addresses that will be used to make connection attempts. Therefore, HEv3 does not iterate on the handling of IPv6 much but rather proposes new rules to optimize the selection of additional protocols with the goal of improving user experience.

As of 2025, IPv6 speed and reliability have significantly improved [3]. Accordingly, IPv6 adoption has continually trended upwards and is approaching almost 50% (see Fig. 1b), as reported by Google [1]. A number of studies, such as [3], [14] and [16] by Papastergiou et al., as well as [17] by G. der Kinderer, show that the focus of the HE concept is gradually shifting more towards optimization in contexts, where a decision between multiple protocols has to be made. This is corroborated by the HEv3 draft. While the underlying purpose of improved user experience is still present, HE is no longer used solely to decide between IPv6 and IPv4.

2. Background

This section provides technical details on each version of the HE protocol. It will summarize their functionality and discuss their differences.

2.1. Happy Eyeballs v1 (RFC 6555)

As the first HE protocol, HEv1 [12] aims to resolve the most harmful problems of dual-stack setups that were present at the time of its release in 2012. It identifies the multiple connection attempts over possibly broken IPv6 as the cause of “significant connection delay compared to an IPv4-only client” [12] and proposes a *Connection Attempt Delay* (CAD) of 150 ms to 250 ms (see Table 1) to combat this behavior.

A dual-stack client trying to establish a connection would first execute a DNS request. If it receives both

an AAAA and an A record it should initiate a connection by sending a SYN segment over IPv6 first. If the client would not receive a SYN/ACK response from the server it would usually attempt further connections using the same IPv6 address [8] causing significant delays. This behavior may be explained by the usage of TCP, which is innately persistent [8]. In contrast, HEv1 proposes a fast fallback, should an IPv6 connection not succeed. This is achieved by immediately sending out a SYN segment over IPv4 after a specific amount of time, which is defined by the CAD. This does not mean that connections, which are already being tried, should be interrupted. Rather, all connections should be raced, with some addresses being prioritized. Only if one connection succeeds, should the other attempts be terminated. According to HEv1, clients must cache this information for about 10 min to avoid having to fall back during subsequent connection attempts. A behavior in cases, where both IPv6 and IPv4 fail and the client received multiple AAAA and/or A records is not yet defined.

In conclusion, HEv1 addresses the biggest problem of dual-stack setups when encountering broken IPv6, that being a significant connection delay, while still staying rather simple and mainly focusing on the fallback to IPv4.

2.2. Happy Eyeballs v2 (RFC 8305)

HEv2 [13] significantly refines the HE algorithm, expanding its functionality to cover the DNS request, among other. It introduces a *Resolution Delay* (RD), that defines how long a client should wait for an AAAA record response after already receiving an A record. If no AAAA record arrives, a connection attempt over IPv4 is initiated (see Fig. 2). HEv2 now also recommends behavior in cases where the DNS request yields multiple records of the same type. To this end, a new parameter, *First Address Family Count* (FAFC), is introduced. It defines how many addresses of the same address family should be tried before falling back to the other one. A value of 1 is recommended but 2 may also be used to favor a particular address family more strongly [13]. This practice is employed only by the Safari browser [14] to further prioritize IPv6. Should the fall back attempt fail, it is recommended to try all the remaining addresses from the favored address family. In cases where these fail as well, connections using the leftover addresses from the other family should be attempted.

Besides the RD, HEv2 introduces additional parameters, which specify the CAD. First established in HEv1

TABLE 1: Comparison of parameters defined for HEv1 [12], v2 [13] and the draft for v3 [15] as of September 2025. Table reproduced from [14]

Parameter	HEv1 (2012)	HEv2 (2017)	HEv3 (2025-ongoing)
Considered protocols	IPv4, IPv6	IPv4, IPv6, DNS	IPv4, IPv6, DNS, QUIC
DNS Records	–	AAAA, A	SVCB, HTTPS, AAAA, A
Resolution Delay	–	50 ms	50 ms
Address selection	IPv6 once, then IPv4	alternating	IP family and L4 protocol
Fixed Conn. Attempt Delay	150 - 250 ms	250 ms	250 ms
Min/Rec./Max when dynamic	–	10 ms / 100 ms / 2 s	10 ms / 100 ms / 2 s

with a recommended value between 150ms and 250ms, now a static value of 250ms is recommended. Other new parameters are a minimum and maximum CAD, with suggested values being 100ms and 2s respectively. The minimum CAD is defined to never be less than 10ms. This helps to avoid cases, where high packet-loss rates cause congestion collapse [13]. Caching the round trip time (RTT) of prior connections allows this data to be used to influence both the ordering of addresses and the CAD, giving priority to those with a low delay.

2.3. Happy Eyeballs v3 (IETF Draft)

While HEv3 [15] is not yet an RFC, it is an IETF draft, in development since 2023. Currently, it does not modify any of the existing parameter values described in HEv2 but extends the functionality to include additional resource records. These are HTTPS and SVCB, which can provide further information, useful in prioritizing the better connection. This includes the protocol, distinguishing between traditional TCP + TLS and HTTP/3 [18] over QUIC [19], the latter being faster and more efficient. Furthermore, services may advertise support for Encrypted Client Hello (ECH) or include ‘address hints’, which supply IP addresses alongside additional routing information, potentially reducing the need for separate A and AAAA lookups. Overall, these newly introduced features allow clients to apply the HE algorithm more dynamically without relying solely on previously cached optimizations, since they are directly provided during DNS resolution.

3. Related Work

Several prior studies have investigated the HE protocol with respect to its effect on the IPv6 landscape, different implementations and their functionality, as well as how HE may be used as a tool for optimization in various situations. The following section will summarize some of them and outline the transformation the HE protocol has undergone in the 13 years since its introduction in 2012.

3.1. Happy Eyeballs as a fix for broken IPv6

Even before the standardization of the first HE version, article [20] by E. Aben was released in 2011. They compare the already existing implementations of different browsers, including Chrome, Firefox and Safari, especially in unison with two versions of Mac OS X. The article reveals that browsers like Chrome had already implemented an HE algorithm, while, e.g. Firefox did

not. Additionally, a harmful behavior called ‘hampering Eyeballs’ is described, which was mostly exhibited by Mac OS X Lion and caused connections to prefer IPv4 even though IPv6 worked perfectly fine. In conclusion, they showed that HE can be a useful tool in improving user experience but might also introduce some noticeable drawbacks if not implemented correctly.

Similarly, article [8] by G. Huston also examines HE implementations as of 2012. Additionally, they first investigate IPv6 brokenness using Windows 7 and an Internet Explorer browser to find concrete delays of about 22s in dual-stack setups. This is corroborated by the findings of Chown and Morse in [11], who examined similar delays using Internet Explorer, as well as delays of up to 45s using browsers like Firefox or Safari. As for the implementations, [8], like [20], also investigates Safari with Mac OS X, Chrome and Firefox. It is shown that both Chrome and Firefox connect to the first address family received by a DNS query, not prioritizing IPv6. Furthermore, Firefox is praised for racing both protocols as soon as their DNS responses arrive, despite this behavior being prone to lead to ‘hampering Eyeballs’ [20]. In contrast, Chrome waits the recommended amount of 300ms before falling back to the other address family. Chrome also displayed longer delays in cases where both protocols failed. Here, the need for new specifications like RD, FAFC and a minimum CAD, which were later introduced in HEv2 [13], is showcased. Lastly, it is discussed that binding too many ports when establishing a TCP connection might be harmful when using Carrier Grade NAT (CGN). In light of this, prioritizing IPv6 with a delay like Chrome’s 300ms is emphasized to be better than a true parallel implementation.

The most recent studies examining HE and IPv6 performance are [2] and [3], both written by Bajpai and Schönwälder, released in 2016 and 2019 respectively. These are long term studies measuring IPv6 adoption, brokenness and latency, as well as the effect of HE since 2013. Since [3] expands on [2], this paper will focus on the former. Similar to [8], they name Teredo [4] and 6to4 [5] as the main cause of IPv6 brokenness and find that it has significantly decreased from 40% in 2009 to 1.9% in 2019. Furthermore, they examine that from 2012 to January 2019, the number of ALEXA 1 M websites that announce AAAA record entries has increased from ~1% to ~19.2% [3]. Additionally, they name Google’s adoption statistic, which has increased from less than 1% in 2012 to ~26% in Jan 2019 [3] and now to almost 50% (see Fig. 1b). Still, 27% of websites with AAAA entries suffer from partial IPv6 failure. Most important

to the functionality of the HE protocol, they find that for 99% of ALEXA 10 K websites (with AAAA entries), an IPv6 connection is preferred 96% of the time [3]. In 81% of these cases, this behavior prefers a slower IPv6 connection, which is almost always only ~ 1 ms behind. They demonstrate that a CAD of 150 ms, compared to the recommended 250 ms, would not significantly impact IPv6 preference. Therefore, they suggest specifying the CAD to a more useful value of 150 ms.

3.2. The state of Happy Eyeballs implementations

While studies like [8] and [20] investigate early HE implementations, the only study measuring these in recent years has been [14] by Sattler et al., using a local testbed, as well as an online tool which can be found under <https://www.happy-eyeballs.net/>. They expanded their investigation to include not only browsers but recursive resolvers and the terminal commands curl [21] and wget [22]. Their findings show that all implementations prefer IPv6 and that all except wget offer a fallback to IPv4. As for the CAD, curl, Firefox and Chromium based browsers use static values of 200 ms, 250 ms, and 300 ms respectively, while Safari employs a dynamic approach, exhibiting CADs ranging from 50 ms up to 2 s. Safari is also shown to be the only browser implementing an RD. It uses the recommended value of 50 ms. This results in both other browser types failing completely should no A record arrive, in spite of a DNS response providing a valid AAAA record. Additionally, only Safari implements its HE algorithm with a FAFC of 2 choosing to more heavily prioritize IPv6. In conclusion, most implementations are still using HEv1 with Safari being the only exception.

3.3. The role of Happy Eyeballs in a modern Internet landscape

As demonstrated by [3], discussed at the end of 3.1, HE is suggested to be further specified with a CAD of 150ms. Additionally, the HEv3 draft [15] incorporates new protocols. Both of these suggestions demonstrate that Happy Eyeballs can be used for optimization. Similarly, reference [16] also discusses how HE can be used to decide between two L4 protocols, these being TCP and SCTP. They find that this approach works, while introducing acceptable additional CPU load for servers, which can be improved with caching. This validates HEv3's idea of using the HE algorithm to not only decide between IPv6 and IPv4, but also different HTTP versions, since HTTP/3 over QUIC uses UDP instead of TCP. Lastly, reference [17] shows how the HE approach can be extended to be used by other protocols, XMPP in this case, with the general goal of improving user experience.

4. Conclusion

The Happy Eyeballs protocol was introduced during a time of prevalent IPv6 issues. Its use as a fix for these problems has been documented to work very well. HE succeeds in significantly improving user experience while not slowing down the transition from IPv4 to IPv6 and is therefore seeing widespread implementation. Still,

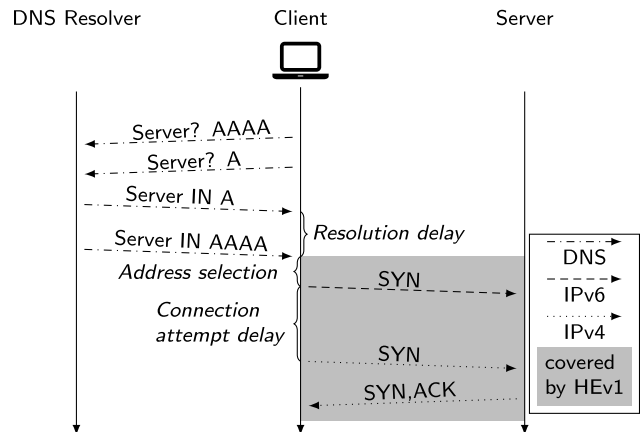


Figure 2: Happy Eyeballs (v2) connection process. Figure reproduced from [14]

HEv1 had to be further specified in HEv2 to more safely guarantee a fallback to IPv4 and optimize the algorithm with concrete values that cover additional cases. For that reason, it is alarming to see that mostly only HEv1 has been implemented, especially since HEv2 secures safe connections to IPv6-only services [14], which are the end goal in the ongoing transition from IPv4.

As of 2025, the state of IPv6 has greatly improved, being used more often with connection times on par with IPv4 most of the time. Furthermore, the internet infrastructure has become increasingly complicated with the introduction of various different protocols. In such an environment HE offers a very powerful approach in selecting between these, being used more for optimization, than only as a tool for consolidation brokenness a specific protocol. Since HE uses a fairly simple idea of racing two options, giving priority to one of them, it could find even more widespread adoption. Future work might further expand on this functionality or integrate this concept into other applications.

In conclusion, Happy Eyeballs was introduced as a fix for broken IPv6 and has developed into an algorithm to select between different protocols, with the option to prioritize some of them.

References

- [1] Geoff Huston. (2025) Google ipv6 statistics. Accessed: 2025-09-21. [Online]. Available: <https://www.google.com/intl/en/ipv6/statistics.html>
- [2] V. Bajpai and J. Schönwälder, “Measuring the effects of happy eyeballs,” in *Proceedings of the 2016 Applied Networking Research Workshop*, ser. ANRW '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 38–44. [Online]. Available: <https://doi.org/10.1145/2959424.2959429>
- [3] V. Bajpai and J. Schönwälder, “A longitudinal view of dual-stacked websites—failures, latency and happy eyeballs,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 577–590, 2019.
- [4] C. Huitema, “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs),” RFC 4380, Feb. 2006. [Online]. Available: <https://www.rfc-editor.org/info/rfc4380>
- [5] B. E. Carpenter and K. Moore, “Connection of IPv6 Domains via IPv4 Clouds,” RFC 3056, Feb. 2001. [Online]. Available: <https://www.rfc-editor.org/info/rfc3056>

- [6] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice, "Evaluating ipv6 adoption in the internet," in *Passive and Active Measurement*, A. Krishnamurthy and B. Plattner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 141–150.
- [7] S. Zander, L. L. Andrew, G. Armitage, G. Huston, and G. Michaelson, "Investigating the ipv6 teredo tunnelling capability and performance of internet clients," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 5, p. 13–20, Sep. 2012. [Online]. Available: <https://doi.org/10.1145/2378956.2378959>
- [8] Google. (2012) Bemused eyeballs: Tailoring dual stack applications for a cgN environment. Accessed: 2025-09-20. [Online]. Available: <https://labs.apnic.net/index.php/2012/05/13/bemused-eyeballs-tailoring-dual-stack-applications-for-a-cgN-environment/>
- [9] (2025) getaddrinfo(3) – linux manual page. man7.org Linux man-pages project. Accessed: 2025-09-21. [Online]. Available: <https://man7.org/linux/man-pages/man3/getaddrinfo.3.html>
- [10] R. P. Draves, "Default Address Selection for Internet Protocol version 6 (IPv6)," RFC 3484, Mar. 2003. [Online]. Available: <https://www.rfc-editor.org/info/rfc3484>
- [11] T. Chown and J. Morse, "Experiences of host behavior in broken ipv6 networks," Presentation at IETF 80, V6OPS Working Group, March 2011, slides–PDF. [Online]. Available: <https://www.ietf.org/proceedings/80/slides/v6ops-12.pdf>
- [12] D. Wing and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts," RFC 6555, Apr. 2012. [Online]. Available: <https://www.rfc-editor.org/info/rfc6555>
- [13] D. Schinazi and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency," RFC 8305, Dec. 2017. [Online]. Available: <https://www.rfc-editor.org/info/rfc8305>
- [14] P. Sattler, M. Kirstein, L. Wüstrich, J. Zirngibl, and G. Carle, "Lazy eye inspection: Capturing the state of happy eyeballs implementations," *arXiv preprint*, vol. arXiv:2412.00263v3, November 2024. [Online]. Available: <https://arxiv.org/abs/2412.00263>
- [15] T. Pauly, D. Schinazi, N. Jaju, and K. Ishibashi, "Happy Eyeballs Version 3: Better Connectivity Using Concurrency," Internet Engineering Task Force, Internet-Draft draft-ietf-happy-happyeyeballs-v3-01, Jul. 2025, work in Progress. [Online]. Available: <https://datatracker.ietf.org/doc/draft-ietf-happy-happyeyeballs-v3/01/>
- [16] G. Papastergiou, K.-J. Grinnemo, A. Brunstrom, D. Ros, M. Tüxen, N. Khademi, and P. Hurtig, "On the cost of using happy eyeballs for transport protocol selection," in *Proceedings of the 2016 Applied Networking Research Workshop*, ser. ANRW '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 45–51. [Online]. Available: <https://doi.org/10.1145/2959424.2959437>
- [17] G. der Kinderen, "Happy eyeballs." [Online]. Available: <https://xmpp.org/extensions/xep-0495.html>
- [18] M. Bishop, "HTTP/3," RFC 9114, Jun. 2022. [Online]. Available: <https://www.rfc-editor.org/info/rfc9114>
- [19] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, May 2021. [Online]. Available: <https://www.rfc-editor.org/info/rfc9000>
- [20] Emile Aben. (2011) Hampering eyeballs - observations on two "happy eyeballs" implementations. Accessed: 2025-09-17. [Online]. Available: <https://labs.ripe.net/author/emileaben/hampering-eyeballs-observations-on-two-happy-eyeballs-implementations/>
- [21] D. Stenberg, "curl," <https://curl.se/>, 2024, retrieved 2025-09-22.
- [22] T. Rühse, D. Shah, and G. Scrivano, "Gnu wget," <https://www.gnu.org/software/wget/>, 2025, retrieved 2025-05-01.