Lazy Eye Inspection: Capturing the State of Happy Eyeballs Implementations

Patrick Sattler, Matthias Kirstein, Lars Wüstrich, **Johannes Zirngibl**, Georg Carle

Friday 26th September, 2025

MIR^3 2025

Technical University of Munich

Max Planck Institute for Informatics

Motivation

Increasing number of network protocols

- IPv6 vs IPv4
- QUIC vs TLS/TCP
- HTTP/3 vs HTTP/2 vs HTTP/1.1
- Encrypted Client Hello vs normal TLS

Clients want the best possible experience

- Best possible protocol stack!
- Any service is better than none!

→ Possible solution: Happy Eyeballs

Motivation

Increasing number of network protocols

- IPv6 vs IPv4
- QUIC vs TLS/TCP
- HTTP/3 vs HTTP/2 vs HTTP/1.1
- Encrypted Client Hello vs normal TLS

Clients want the best possible experience

- Best possible protocol stack!
- Any service is better than none!

→ Possible solution: Happy Eyeballs

Motivation

Increasing number of network protocols

- IPv6 vs IPv4
- QUIC vs TLS/TCP
- HTTP/3 vs HTTP/2 vs HTTP/1.1
- Encrypted Client Hello vs normal TLS

Clients want the best possible experience

- Best possible protocol stack!
- Any service is better than none!

→ Possible solution: Happy Eyeballs

Research Questions

How can we measure the Happy Eyeballs behavior?

What is the current behavior of clients?

What influences the client behavior?

Research Questions

How can we measure the Happy Eyeballs behavior?

What is the current behavior of clients?

What influences the client behavior?

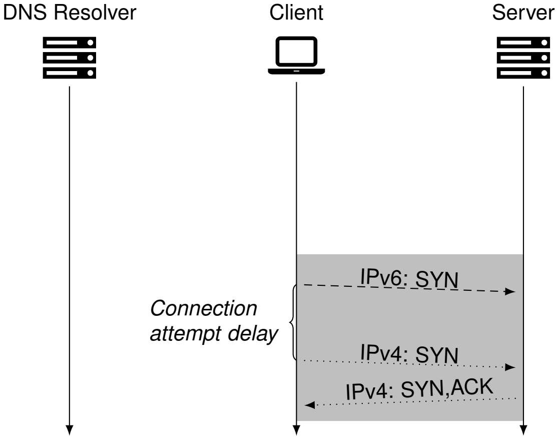
Research Questions

How can we measure the Happy Eyeballs behavior?

What is the current behavior of clients?

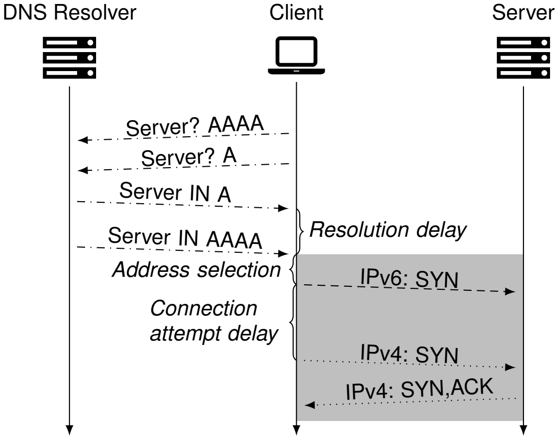
What influences the client behavior?

Happy Eyeballs



- Version 1:
  - RFC 6555, 2012
  - Guidelines for address selection
  - Guidelines for connection establishments and delays
- Version 2:
  - RFC 8305, 2017
  - Guidelines for DNS
  - Extended guidelines for address selection

Happy Eyeballs



- Version 1:
  - RFC 6555, 2012
  - Guidelines for address selection
  - Guidelines for connection establishments and delays
- Version 2:
  - RFC 8305, 2017
  - Guidelines for DNS
  - Extended guidelines for address selection

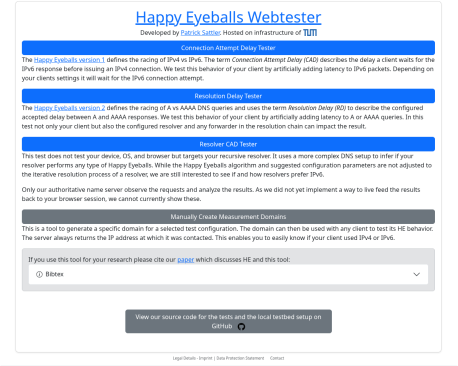
Happy Eyeballs

Parameter	HEv1 (2012)	HEv2 (2017)
Considered protocols	IPv4, IPv6	IPv4, IPv6, DNS
DNS Records	-	AAAA, A
Resolution Delay	-	50 ms
Address selection	IPv6 once, then IPv4	alternating
Fixed Conn. Attempt Delay	150-250 ms	250 ms
↦Min/Rec./Max when dynamic	-	10 ms / 100 ms / 2 s

Web-based Testing Tool

Controlled test environment:

- `https://www.happy-eyeballs.net/`
- Domains encode test behavior and trigger different behavior
- Connection attempt delay:
  - Added delay based on tc-netem
  - Different IP addresses assigned to specific delays
- Resolution Delay:
  - Custom authoritative nameserver
  - Delays responses based on queried domain



Happy Eyeballs Webtester

Developed by Patrick Sattler. Hosted on infrastructure of TUM

Connection Attempt Delay Tester

The Happy Eyeballs version 1 defines the racing of IPv4 vs IPv6. The term *Connection Attempt Delay (CAD)* describes the delay a client waits for the IPv6 response before issuing an IPv4 connection. We test this behavior of your client by artificially adding latency to IPv6 packets. Depending on your clients settings it will wait for the IPv6 connection attempt.

Resolution Delay Tester

The Happy Eyeballs version 2 defines the racing of A vs AAAA DNS queries and uses the term *Resolution Delay (RD)* to describe the configured accepted delay between A and AAAA responses. We test this behavior of your client by artificially adding latency to A or AAAA queries. In this test not only your client but also the configured resolver and any forwarder in the resolution chain can impact the result.

Resolver CAD Tester

This test does not test your device, OS, and browser but targets your recursive resolver. It uses a more complex DNS setup to infer if your resolver performs any type of Happy Eyeballs. While the Happy Eyeballs algorithm and suggested configuration parameters are not adjusted to the iterative resolution process of a resolver, we are still interested to see if and how resolvers prefer IPv6.

Only our authoritative name server observe the requests and analyze the results. As we did not yet implement a way to live feed the results back to your browser session, we cannot currently show these.

Manually Create Measurement Domains

This is a tool to generate a specific domain for a selected test configuration. The domain can then be used with any client to test its HE behavior. The server always returns the IP address at which it was contacted. This enables you to easily know if your client used IPv4 or IPv6.

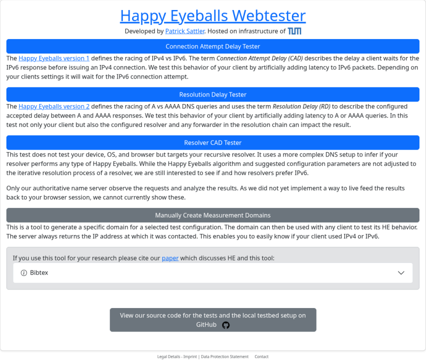
If you use this tool for your research please cite our paper which discusses HE and this tool:

ⓘ Bibtex
View our source code for the tests and the local testbed setup on GitHub
Legal Details - Imprint | Data Protection Statement Contact

Web-based Testing Tool

Controlled test environment:

- `https://www.happy-eyeballs.net/`
- Domains encode test behavior and trigger different behavior
- Connection attempt delay:
  - Added delay based on tc-netem
  - Different IP addresses assigned to specific delays
- Resolution Delay:
  - Custom authoritative nameserver
  - Delays responses based on queried domain



Happy Eyeballs Webtester

Developed by Patrick Sattler. Hosted on infrastructure of TUM

Connection Attempt Delay Tester

The Happy Eyeballs version 1 defines the racing of IPv4 vs IPv6. The term *Connection Attempt Delay (CAD)* describes the delay a client waits for the IPv6 response before issuing an IPv4 connection. We test this behavior of your client by artificially adding latency to IPv6 packets. Depending on your clients settings it will wait for the IPv6 connection attempt.

Resolution Delay Tester

The Happy Eyeballs version 2 defines the racing of A vs AAAA DNS queries and uses the term *Resolution Delay (RD)* to describe the configured accepted delay between A and AAAA responses. We test this behavior of your client by artificially adding latency to A or AAAA queries. In this test not only your client but also the configured resolver and any forwarder in the resolution chain can impact the result.

Resolver CAD Tester

This test does not test your device, OS, and browser but targets your recursive resolver. It uses a more complex DNS setup to infer if your resolver performs any type of Happy Eyeballs. While the Happy Eyeballs algorithm and suggested configuration parameters are not adjusted to the iterative resolution process of a resolver, we are still interested to see if and how resolvers prefer IPv6.

Only our authoritative name server observe the requests and analyze the results. As we did not yet implement a way to live feed the results back to your browser session, we cannot currently show these.

Manually Create Measurement Domains

This is a tool to generate a specific domain for a selected test configuration. The domain can then be used with any client to test its HE behavior. The server always returns the IP address at which it was contacted. This enables you to easily know if your client used IPv4 or IPv6.

If you use this tool for your research please cite our paper which discusses HE and this tool:

ⓘ Bibtex

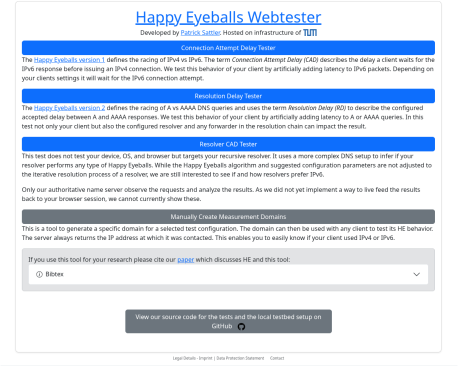
View our source code for the tests and the local testbed setup on GitHub

Legal Details - Imprint | Data Protection Statement Contact

Web-based Testing Tool

Controlled test environment:

- `https://www.happy-eyeballs.net/`
- Domains encode test behavior and trigger different behavior
- Connection attempt delay:
  - Added delay based on tc-netem
  - Different IP addresses assigned to specific delays
- Resolution Delay:
  - Custom authoritative nameserver
  - Delays responses based on queried domain



Happy Eyeballs Webtester

Developed by Patrick Sattler. Hosted on infrastructure of TUM

Connection Attempt Delay Tester

The Happy Eyeballs version 1 defines the racing of IPv4 vs IPv6. The term *Connection Attempt Delay (CAD)* describes the delay a client waits for the IPv6 response before issuing an IPv4 connection. We test this behavior of your client by artificially adding latency to IPv6 packets. Depending on your clients settings it will wait for the IPv6 connection attempt.

Resolution Delay Tester

The Happy Eyeballs version 2 defines the racing of A vs AAAA DNS queries and uses the term *Resolution Delay (RD)* to describe the configured accepted delay between A and AAAA responses. We test this behavior of your client by artificially adding latency to A or AAAA queries. In this test not only your client but also the configured resolver and any forwarder in the resolution chain can impact the result.

Resolver CAD Tester

This test does not test your device, OS, and browser but targets your recursive resolver. It uses a more complex DNS setup to infer if your resolver performs any type of Happy Eyeballs. While the Happy Eyeballs algorithm and suggested configuration parameters are not adjusted to the iterative resolution process of a resolver, we are still interested to see if and how resolvers prefer IPv6.

Only our authoritative name server observe the requests and analyze the results. As we did not yet implement a way to live feed the results back to your browser session, we cannot currently show these.

Manually Create Measurement Domains

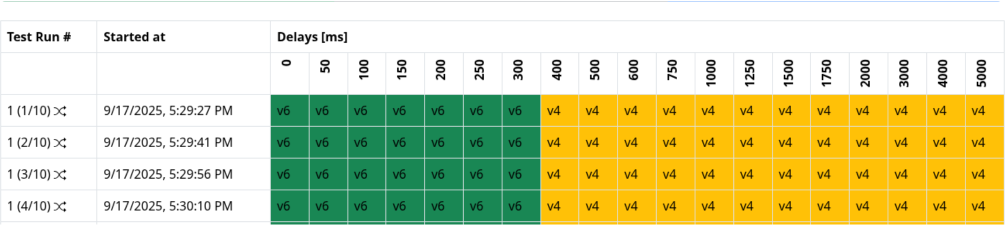
This is a tool to generate a specific domain for a selected test configuration. The domain can then be used with any client to test its HE behavior. The server always returns the IP address at which it was contacted. This enables you to easily know if your client used IPv4 or IPv6.

If you use this tool for your research please cite our paper which discusses HE and this tool:

View our source code for the tests and the local testbed setup on GitHub

Connection Attempt Delay

Chrome

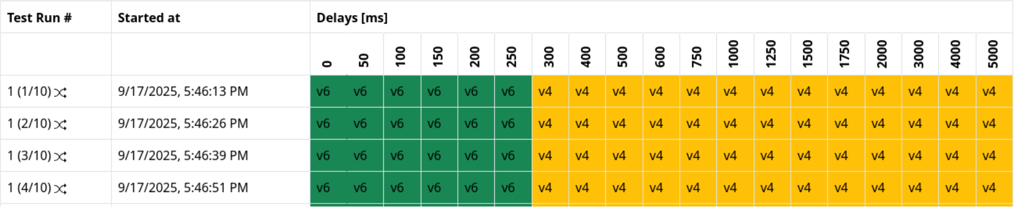


Test Run #	Started at	Delays [ms]																	
		0	50	100	150	200	250	300	400	500	600	750	1000	1250	1500	1750	2000	3000	4000	5000
1 (1/10) 🔀	9/17/2025, 5:29:27 PM	v6	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (2/10) 🔀	9/17/2025, 5:29:41 PM	v6	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (3/10) 🔀	9/17/2025, 5:29:56 PM	v6	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (4/10) 🔀	9/17/2025, 5:30:10 PM	v6	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4

- Attempts an IPv6 connection first
- Static connection attempt delay of 300ms

Connection Attempt Delay

Firefox

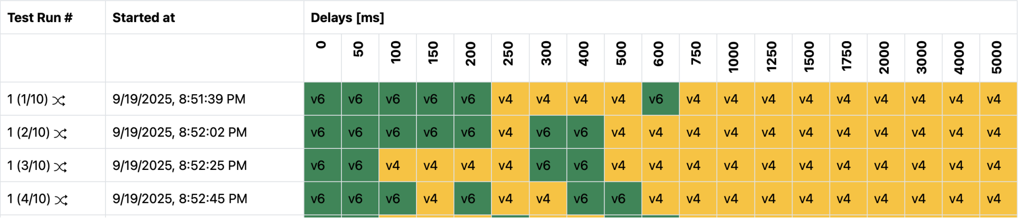


Test Run #	Started at	Delays [ms]																		
		0	50	100	150	200	250	300	400	500	600	750	1000	1250	1500	1750	2000	3000	4000	5000
1 (1/10) 🔀	9/17/2025, 5:46:13 PM	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (2/10) 🔀	9/17/2025, 5:46:26 PM	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (3/10) 🔀	9/17/2025, 5:46:39 PM	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (4/10) 🔀	9/17/2025, 5:46:51 PM	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4

- Attempts an IPv6 connection first
- Static connection attempt delay of 300ms

Connection Attempt Delay

Safari

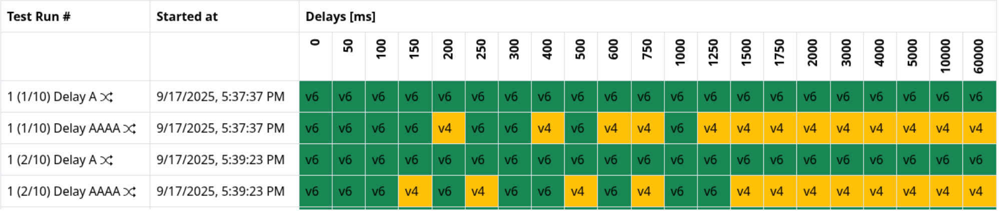


Test Run #	Started at	Delays [ms]																		
		0	50	100	150	200	250	300	400	500	600	750	1000	1250	1500	1750	2000	3000	4000	5000
1 (1/10) 🔀	9/19/2025, 8:51:39 PM	v6	v6	v6	v6	v6	v4	v4	v4	v4	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (2/10) 🔀	9/19/2025, 8:52:02 PM	v6	v6	v6	v6	v6	v4	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (3/10) 🔀	9/19/2025, 8:52:25 PM	v6	v6	v4	v4	v4	v4	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (4/10) 🔀	9/19/2025, 8:52:45 PM	v6	v6	v6	v4	v6	v4	v4	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4

- Attempts an IPv6 connection first
- Dynamic connection attempt delay

Resolution Delay

Chrome

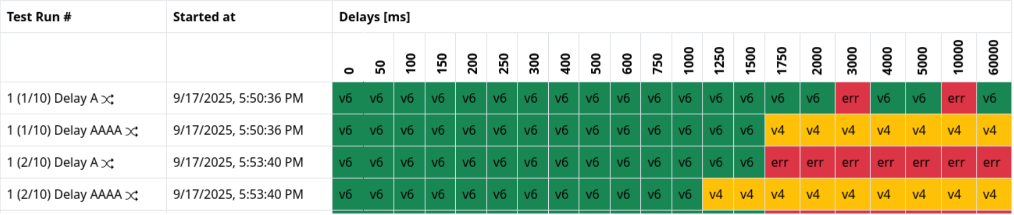


Test Run #	Started at	Delays [ms]																				
		0	50	100	150	200	250	300	400	500	600	750	1000	1250	1500	1750	2000	3000	4000	5000	10000	60000
1 (1/10) Delay A ⤮	9/17/2025, 5:37:37 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6
1 (1/10) Delay AAAA ⤮	9/17/2025, 5:37:37 PM	v6	v6	v6	v6	v4	v6	v6	v4	v6	v4	v4	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (2/10) Delay A ⤮	9/17/2025, 5:39:23 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6
1 (2/10) Delay AAAA ⤮	9/17/2025, 5:39:23 PM	v6	v6	v6	v4	v6	v4	v6	v6	v4	v6	v4	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4

- No resolution delay
- Process waits for both records
- Non-deterministic timeout during resolution

Resolution Delay

Firefox

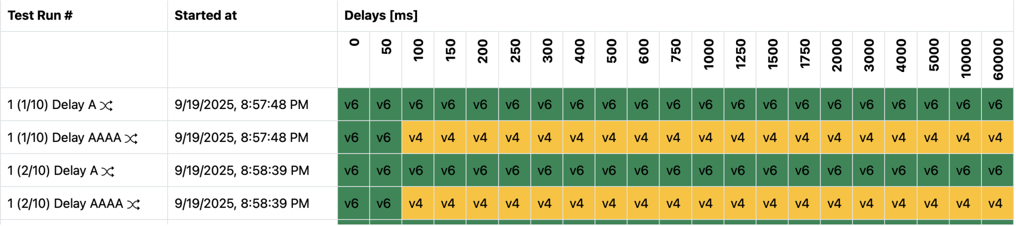


Test Run #	Started at	Delays [ms]																				
		0	50	100	150	200	250	300	400	500	600	750	1000	1250	1500	1750	2000	3000	4000	5000	10000	60000
1 (1/10) Delay A 🔀	9/17/2025, 5:50:36 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	err	v6	v6	err	v6
1 (1/10) Delay AAAA 🔀	9/17/2025, 5:50:36 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4
1 (2/10) Delay A 🔀	9/17/2025, 5:53:40 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	err	err	err	err	err	err	err
1 (2/10) Delay AAAA 🔀	9/17/2025, 5:53:40 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4

- No resolution delay
- Process waits for both Records
- Connections time out if no A record is available

Resolution Delay

Safari



Test Run #	Started at	Delays [ms]																				
		0	50	100	150	200	250	300	400	500	600	750	1000	1250	1500	1750	2000	3000	4000	5000	10000	60000
1 (1/10) Delay A 🔀	9/19/2025, 8:57:48 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6
1 (1/10) Delay AAAA 🔀	9/19/2025, 8:57:48 PM	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4
1 (2/10) Delay A 🔀	9/19/2025, 8:58:39 PM	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6	v6
1 (2/10) Delay AAAA 🔀	9/19/2025, 8:58:39 PM	v6	v6	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4	v4

- Implements a resolution delay of 50 ms

One Browser, One Behavior?

The behavior does not only depend on the browser:

- Most browsers do not implement domain resolution in their stable version at the moment
  - The resolution delay is hard to implement
  - In some cases, a delayed A record leads to connection errors in general
- DNS resolvers have different cache or timeout behavior
  - Large delays might lead to timeouts
  - Resolvers might implement HE themselves
- Proxy/tunnel setups influence the behaviors
  - e.g., Akamai and Cloudflare show different behavior used as iCloud Private Relay egress proxies

One Browser, One Behavior?

The behavior does not only depend on the browser:

- Most browsers do not implement domain resolution in their stable version at the moment
  - The resolution delay is hard to implement
  - In some cases, a delayed A record leads to connection errors in general
- DNS resolvers have different cache or timeout behavior
  - Large delays might lead to timeouts
  - Resolvers might implement HE themselves
- Proxy/tunnel setups influence the behaviors
  - e.g., Akamai and Cloudflare show different behavior used as iCloud Private Relay egress proxies

One Browser, One Behavior?

The behavior does not only depend on the browser:

- Most browsers do not implement domain resolution in their stable version at the moment
  - The resolution delay is hard to implement
  - In some cases, a delayed A record leads to connection errors in general
- DNS resolvers have different cache or timeout behavior
  - Large delays might lead to timeouts
  - Resolvers might implement HE themselves
- Proxy/tunnel setups influence the behaviors
  - e.g., Akamai and Cloudflare show different behavior used as iCloud Private Relay egress proxies

HAPPY

Heuristics and Algorithms to Prioritize Protocol deploYment

- New working group at the IETF
- Works on Happy Eyeballs v3
- "connection establishment that starts with an FQDN and ends with a single established connection to a server"

Parameter	HEv1 (2012)	HEv2 (2017)	HEv3 (2025-ongoing)
Considered protocols	IPv4, IPv6	IPv4, IPv6, DNS	IPv4, IPv6, DNS, QUIC, ECH
DNS Records	-	AAAA, A	SVCB, HTTPS, AAAA, A
Resolution Delay	-	50 ms	50 ms
Address selection	IPv6 once, then IPv4	alternating	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

HAPPY

Heuristics and Algorithms to Prioritize Protocol deploYment

- New working group at the IETF
- Works on Happy Eyeballs v3
- "connection establishment that starts with an FQDN and ends with a single established connection to a server"

Parameter	HEv1 (2012)	HEv2 (2017)	HEv3 (2025-ongoing)
Considered protocols	IPv4, IPv6	IPv4, IPv6, DNS	IPv4, IPv6, DNS, QUIC, ECH
DNS Records	-	AAAA, A	SVCB, HTTPS, AAAA, A
Resolution Delay	-	50 ms	50 ms
Address selection	IPv6 once, then IPv4	alternating	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

HAPPY

Heuristics and Algorithms to Prioritize Protocol deploYment

- New working group at the IETF
- Works on Happy Eyeballs v3
- "connection establishment that starts with an FQDN and ends with a single established connection to a server"

Parameter	HEv1 (2012)	HEv2 (2017)	HEv3 (2025-ongoing)
Considered protocols	IPv4, IPv6	IPv4, IPv6, DNS	IPv4, IPv6, DNS, QUIC, ECH
DNS Records	-	AAAA, A	SVCB, HTTPS, AAAA, A
Resolution Delay	-	50 ms	50 ms
Address selection	IPv6 once, then IPv4	alternating	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

Happy Eyeballs v3

Asynchronous 3-step process

- Asynchronous resolution
  - Switch to sorting if *good* set is available or resolution delay triggers
  - Accept delayed results and update list available addresses
- Address sorting
  - Sort addresses in different groups (e.g., ECH, QUIC)
  - Update sorting if new information is available
  - Consider cache per group/target
- Connection establishment
  - Iterate through sorted list with a CAD

Drastic increase of complexity

Happy Eyeballs v3

Asynchronous 3-step process

- Asynchronous resolution
  - Switch to sorting if *good* set is available or resolution delay triggers
  - Accept delayed results and update list available addresses
- Address sorting
  - Sort addresses in different groups (e.g., ECH, QUIC)
  - Update sorting if new information is available
  - Consider cache per group/target
- Connection establishment
  - Iterate through sorted list with a CAD

Drastic increase of complexity

Happy Eyeballs v3

Asynchronous 3-step process

- Asynchronous resolution
  - Switch to sorting if *good* set is available or resolution delay triggers
  - Accept delayed results and update list available addresses
- Address sorting
  - Sort addresses in different groups (e.g., ECH, QUIC)
  - Update sorting if new information is available
  - Consider cache per group/target
- Connection establishment
  - Iterate through sorted list with a CAD

Drastic increase of complexity

Happy Eyeballs v3

Asynchronous 3-step process

- Asynchronous resolution
  - Switch to sorting if *good* set is available or resolution delay triggers
  - Accept delayed results and update list available addresses
- Address sorting
  - Sort addresses in different groups (e.g., ECH, QUIC)
  - Update sorting if new information is available
  - Consider cache per group/target
- Connection establishment
  - Iterate through sorted list with a CAD

Drastic increase of complexity

Happy Eyeballs v3

Asynchronous 3-step process

- Asynchronous resolution
  - Switch to sorting if *good* set is available or resolution delay triggers
  - Accept delayed results and update list available addresses
- Address sorting
  - Sort addresses in different groups (e.g., ECH, QUIC)
  - Update sorting if new information is available
  - Consider cache per group/target
- Connection establishment
  - Iterate through sorted list with a CAD

Drastic increase of complexity

Happy Eyeballs v3

Asynchronous 3-step process

- Asynchronous resolution
  - Switch to sorting if *good* set is available or resolution delay triggers
  - Accept delayed results and update list available addresses
- Address sorting
  - Sort addresses in different groups (e.g., ECH, QUIC)
  - Update sorting if new information is available
  - Consider cache per group/target
- Connection establishment
  - Iterate through sorted list with a CAD

Drastic increase of complexity

Conclusion

MAX PLANCK INSTITUTE
FOR INFORMATICS

Happy Eyeballs:

- initially focused on IPv6 vs IPv4
- Stagger connections attempts
- Version 2 includes DNS

How can we measure the Happy Eyeballs behavior?

- Our tool allows HEv1 and HEv2 tests
- Updates to come soon

What is the current behavior of clients?

- Browsers support HE, but mostly HEv1

What influences the client behavior?

- DNS resolvers and proxies are impacting the behavior and future evaluations

Paper:

happy-eyeballs.net:

Conclusion

Happy Eyeballs:

- initially focused on IPv6 vs IPv4
- Stagger connections attempts
- Version 2 includes DNS

How can we measure the Happy Eyeballs behavior?

- Our tool allows HEv1 and HEv2 tests
- Updates to come soon

What is the current behavior of clients?

- Browsers support HE, but mostly HEv1

What influences the client behavior?

- DNS resolvers and proxies are impacting the behavior and future evaluations

Paper:

happy-eyeballs.net:

What comes next?

Extend the Happy Eyeballs Tool:

- Deploy further vantage points to reduce network impact
- Customize measurements for power users
- Add HEv3 functionality
  - Add SVCB/HTTPS resource records
  - Add QUIC/H3/ECH support
  - Extend configurations

Paper:

happy-eyeballs.net: