

Happy Eyeballs Implementation Report

Simon Perreault

Viagénie

simon.perreault@viagenie.ca

<http://www.viagenie.ca>

Happy Eyeballs



- draft-ietf-v6ops-happy-eyeballs
- Most current applications try IPv6 first and fall back on IPv4 after a timeout.
- Lookup+connect to IPv4 and IPv6 in parallel. Keep the connection that “wins”, drop the other one.
- “P” state variable that delays IPv4 or IPv6 based on previous connection attempts.
 - e.g. P increases as IPv6 “wins” --> IPv4 is delayed --> IPv4 becomes rarely used
 - Goal: prevent SYN-spamming
 - Global and per-destination.

Implementation



- Language: Erlang (Ericsson's high-level functional language for concurrent programming)
- Goal: Make it easy to test and play with algorithms.
- Download URL:
<http://www.viagenie.ca/ipv6/he-0.1.tar.gz>

Findings



- Global “P” mechanism is broken
 - The Internet is dominated by single-stack IPvX hosts. Therefore, IPvY will rarely win. P will tend to favour IPvX, and IPvY will never be used for dual-stack hosts.
 - Current situation: $X = 4$, $Y = 6$
 - This means that IPv6 would be disadvantaged today.
 - Ideas for fixing:
 - Only update P for dual-stack hosts. Need to wait for the lookup to finish before updating. (Do it asynchronously.)
 - Only use per-destination P.
 - Kill P: Do not try to avoid SYN-spamming.

Findings



- Avoid useless delays
 - a)
 - Spec says: 1) wait 2) lookup 3) connect.
 - Better: If lookup for IPvX returns no result, IPvY should stop waiting and start lookup immediately.
 - b)
 - Goal of P is to avoid SYN-spamming. DNS spamming is considered OK.
 - Therefore steps should be 1) lookup 2) wait 3) connect

Future work



- Merge into an Erlang HTTP proxy
 - Makes all browsers HE-enabled instantaneously!
- A real OS-level implementation in C
 - `int connectByName(const char *host);`
 - Not as easy as it looks.
 - Cannot be properly implemented in terms of `getaddrinfo()`.
 - Some contexts need threads (something like the above would be OK), other contexts need non-blocking sockets (completely different API needed).
 - “The IETF doesn't do APIs.”
 - The biggest road block is the lack of an OS-level API.