Happy Eyeballs: Success with Dual-Stack Hosts

Andrew Yourtchenko, ayourtch@cisco.com
PROBLEM AND HAPPY EYEBALLS SOLUTION
The dark reality of IPv4 exhaustion

Pour your last IPv4 /24 here

Push up here to get one 5-tuple with a public address
Wasn’t there a better solution?
Implementation details...
Dualstack connection sequence

1) Getaddrinfo(): hostname => address list

2) Try the addresses sequentially
Problem and Solution

• Dual-stack client connecting to dual-stack server
• Dual-stack cannot be slower than IPv4
• If slower, users blame IPv6 and disable IPv6!

• IPv6 cannot be slower than IPv4
The Happy Eyeballs Solution

Let the quickest win

1. Foo.com
   I want a connection to foo.com

2. A?
   Wait for the reply

3. AAAAA?
   Wait for the reply

4. connect
   Success?

5. connect
   Success?

6. Success!
   Get webpage
Optimizing Happy Eyeballs

The winner takes it all

Delay the slow one

Demote on failure
RFC 6555: Happy Eyeballs

• Users are happy – fast response even if IPv6 (or IPv4) path is down
• Network administrators are happy
  – Users no longer trying to disable IPv6
  – Reduces IPv4 usage (reduces load on CGN)
• Content providers are happy
  – Improved geolocation and DoS visibility with IPv6
IMPLEMENTATIONS
Happy Eyeballs Coverage

- Web browsing is *the* most common application
  
  [http://jazzychad.net/dcpu.html](http://jazzychad.net/dcpu.html)

- First, improve the web browsing experience
- Second, improve other applications
  - Instant messaging, email client, etc.
Implementations

- Google Chrome (in current stable channel)
- Mozilla Firefox version 10

- Apple OSX 10.7 ("Lion")
  - getaddrinfo()
  - Safari
- Apple iOS 4.3.1
Chrome and Firefox Implementation

• Utilizes long-established 250-300ms ‘backup’ thread
  – Originally just tried the next **IP address**
  – Happy Eyeballs: tries the next **IP address family**

• Follows getaddrinfo() address preference
  – IPv6 is usually preferred by the Operating System

• Result: IPv6 gets 250-300ms head start
Apple Implementation

• Apple Framework calling CFSocketStream
  – A and AAAA queried simultaneously
  – Attempt connection immediately
  – First to connect “wins”

• “Legacy” applications calling getaddrinfo()
  – Addresses sorted based on previous connection success and connection failure

• Result: user connects to fastest of IPv6 or IPv4

Happy Eyeballs
TROUBLESHOOTING
Troubleshooting

• IPv{4/6} outages are not obvious to users
  – To the user, things “just work”
  – Network administrator doesn’t get complaint
Troubleshooting

• Immediate IPv4 traffic when IPv6 is slow
  – Complicates NAT44 scaling
Troubleshooting

• Conclusion: Network tools need to **actively** monitor IPv6 and IPv4 quality
  – Active monitoring should be considered
FUTURE WORK
Beyond Web Browsers

• If users are waiting, need Happy Eyeballs
• Voice over IP has Happy algorithm
  – SIP: RFC6157
• Happy Eyeballs in Applications or OS, or both?
  – Email, Instant Messaging, ssh, ...
  – Games
  – Linux, FreeBSD, OS X, Windows, ...
Future Work

• Happy Eyeballs uses connection setup time
• Future work:
  – Throughput (streaming video)
  – Jitter/Loss (interactive audio/video)
  – Path MTU (9000 byte MTU)
  – Multipath TCP (simultaneous connections)
  – Non-TCP transport protocol (SCTP)
SUMMARY
Happy Eyeballs

- Happy users
  - Fast connections to servers

- Happy network administrators
  - Users won’t disable IPv6
  - Less load on CGN

- Happy content providers
  - Fast connection to servers
  - Better location & DoS visibility with IPv6
Questions


Andrew Yourtchenko,
ayourtch@cisco.com