Benchmarking Methodology F. Baker
Internet-Draft Cisco Systems
Intended status: Informational August 15, 2011
Expires: February 16, 2012

Testing Eyeball Happiness
draft-baker-bmwg-testing-eyeball-happiness-05

Abstract

The amount of time it takes to establish a session using common transport APIs in dual stack networks and networks with filtering such as proposed in BCP 38 is a barrier to IPv6 deployment. This note describes a test that can be used to determine whether an application can reliably establish sessions quickly in a complex environment such as dual stack (IPv4+IPv6) deployment or IPv6 deployment with multiple prefixes and upstream ingress filtering. This test is not a test of a specific algorithm, but of the external behavior of the system as a black box. Any algorithm that has the intended external behavior will be accepted by it.

Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The Happy Eyeballs [I-D.ietf-v6ops-happy-eyeballs] specification notes an issue in deployed multi-prefix IPv6-only and dual stack networks, and proposes a correction. [RFC5461] similarly looks at TCP’s response to so-called "soft errors" (ICMP host and network unreachable messages), pointing out an issue and a set of possible solutions.

In a dual stack network (i.e., one that contains both IPv4 [RFC0791] and IPv6 [RFC2460] prefixes and routes), or in an IPv6-only network that uses multiple prefixes allocated by upstream providers that implement BCP 38 Ingress Filtering [RFC2827], the fact that two hosts that need to communicate have addresses using the same architecture does not imply that the network has usable routes connecting them, or that those addresses are useful to the applications in question. In addition, the process of establishing a session using the Sockets API [RFC3493] is generally described in terms of obtaining a list of possible addresses for a peer (which will normally include both IPv4 and IPv6 addresses) using getaddrinfo() and trying them in sequence until one succeeds or all have failed. This naive algorithm, if implemented as described, has the side-effect of making the worst case delay in establishing a session far longer than human patience normally allows.

This has the effect of discouraging users from enabling IPv6 in their equipment, or content providers from offering AAAA records for their services.

This note describes a test to determine how quickly an application can reliably open sessions in a complex environment, such as dual stack (IPv4+IPv6) deployment or IPv6 deployment with multiple prefixes and upstream ingress filtering. This is not a test of a specific algorithm, but a measurement of the external behavior of the application and its host system as a black box. The "happy eyeballs" question is this: how long does it take an application to open a session with a server or peer, under best case and worst case conditions?

The methods defined here make the assumption that the initial communication set-up of many applications can be summarized by the measuring the DNS query/response and transport layer handshaking, because no application-layer communication takes place without these steps.

The methods and metrics defined in this note are ideally suited for Laboratory operation, as this affords the greatest degree of control to modify configurations quickly and produce consistent results.

However, if the device under test is operated as a single user with limited query and stream generation, then there’s no concern about overloading production network devices with a single "set of eyeballs". Therefore, these procedures and metrics MAY be applicable to production network application, as long as the injected traffic represents a single user’s typical traffic load, and the testers adhere to the precautions of the relevant network with respect to re-configuration of devices in production.

2. Measuring Eyeball Happiness

This measurement determines the amount of time it takes an application to establish a session with a peer in the presence of at least one IPv4 and multiple IPv6 prefixes and a variety of network behaviors. ISPs are reporting that a host (MacOSX, Windows, Linux, FreeBSD, etc) that has more than one address (an IPv4 and an IPv6 address, two global IPv6 addresses, etc) may serially try addresses, allowing each TCP setup to expire, taking several seconds for each attempt. There have been reports of lengthy session setup times - in various application and OS combinations anywhere from multi-second to half an hour - as a result. The amount of time necessary to establish communication between two entities should be approximately the same regardless of the type of address chosen or the viability of routing in the specific network; users will expect this time to be consistent with their current experience (else, happiness is at risk).

2.1. Happy Eyeballs test bed configuration

The configuration of equipment and applications is as shown in Figure 1.

```
+--------+ |                       |198.51.100.0/24
|Protocol| |192.0.2.0/24           |2001:DB8:0:2::/64
|Analyzer+-+2001:DB8:1:0::/64      |2001:DB8:1:4::/64
+--------+ |2001:DB8:0:1::/64      |2001:DB8:2:4::/64
           |                       |
   +-----+ |                       | +-----+
   |Alice+-+                       +-+ Bob |
   +-----+ | +-------+  +-------+ | +-----+
           +-+Router1|  |Router2+-+
   +-----+ | +-----+-+  +-+-----+ |
   | DNS +-+       |      |       |
   +-----+ |      -+------+-      |
           |    203.0.113.0/24    |
           |    2001:DB8:0:3::/64 |
```

Figure 1: Generic Test Environment

Alice is the unit being measured, the computer running the process that will establish a session with Bob for the application in question. DNS is represented in the diagram as a separate system, as is the protocol analyzer that will watch Alice's traffic. This is not absolutely necessary; If one computer can run tcpdump and a DNS server process - and for that matter subsume the routers - that is acceptable. The units are separated in the test for purposes of clarity.

On each test run, configuration is performed in Router 1 to permit only one route to work. There are various ways this can be accomplished, including but not limited to installing

- o a filter that drops datagrams to Bob resulting in an ICMP "administratively prohibited",

- o a filter that silently drops datagrams to Bob,

- o a null route or removing the route to one of Bob's prefixes, resulting in an ICMP "destination unreachable", and

- o a middleware program that responds with a TCP RST.

- o Path MTU issues

The Path MTU Discovery [RFC1191][RFC1981] matter requires some explanation. With IPv6, and with IPv4 when "Do Not Fragment" is set, a router with a message too large for an interface discards it and replies with an ICMPv4 "Destination Unreachable: Datagram Too Big" or ICMPv6 "Packet Too Big". If this packet is lost, the source doesn't know what size to fragment to and has no indication that fragmentation is required. A configuration for this scenario would set the MTU on 203.0.113.0/24 or 2001:DB8:0:3::/64 to the smallest allowed by the address family (576 or 1280) and disable generation of the indicated ICMP message. Note that [RFC4821] is intended to address these issues.

The tester should try different methods to determine whether differences in this configuration make a difference in the test. For example, one might find that the application under test responds differently to a TCP RST than to a silent packet loss. Each of these scenarios should be tested; if doing so is too difficult, the most important is the silent packet loss case, as it is the worst case.

2.2. Happy Eyeballs test procedure

Consider a network as described in Section 2.1. Alice and Bob each have a set of one or more IPv4 and two or more IPv6 addresses. Bob's are in DNS, where Alice can find them; Alice's and others may be there as well, but are not relevant to the test. Routers 1 and 2 are configured to route the relevant prefixes. Different measurement trials revise an access list or null route in Router 1 that would prevent traffic Alice->Bob using each of Bob's addresses. If Bob has a total of N addresses, we run the measurement at least N times, permitting exactly one of the addresses to enjoy end to end communication each time. If the DNS service randomizes the order of the addresses, this may not result in a test requiring establishment of a connection to all of the addresses; in this case, the test will have to be run repeatedly until in at least one instance a TCP SYN or its equivalent is seen for each relevant address. The tester should either flush the resolver cache between iterations, to force repeated DNS resolution, or should wait for at least the DNS RR TTL on each resource record. In the latter case, the tester should also observe DNS re-resolving; if not, the application is not correctly using DNS.

This specification assumes common LAN technology with no competing traffic and nominal propagation delays, so that they are not a factor in the measurement.

The objective is to measure the amount of time required to establish a session. This includes the time from Alice's initial DNS request through one or more attempts to establish a session to the session being established, as seen in the LAN trace. The simplest way to measure this will be to put a traffic analyzer on Alice's point of attachment and capture the messages exchanged by Alice.

```
 DNS Server                    Alice                    Bob
     |                           |                        |
  1. |<--www.example.com A------|                        |
  2. |<--www.example.com AAAA---|                        |
  3. |---198.51.100.1---------->|                        |
  4. |---2001:DB8:0:2::1------->|                        |
  5. |                           |                        |
  6. |                           |--TCP SYN, IPv6--->X   |<***********
  7. |                           |--TCP SYN, IPv6--->X   |      |
  8. |                           |--TCP SYN, IPv6--->X   | TCP 3wHS
  9. |                           |                        |   Time
 10. |                           |--TCP SYN, IPv4------->|(any family)
 11. |                           |<-TCP SYN+ACK, IPv4----|      |
 12. |                           |--TCP ACK, IPv4------->|<***********
```

Figure 2: Message flow using TCP

In a TCP-based application (Figure 2), that would be from the DNS request on line 1 through the first completion of a TCP three-way handshake, the transmission on line 12.

```
 DNS Server                   Alice                    Bob
      |                          |                       |
  1.  |<--www.example.com A------|                       |
  2.  |<--www.example.com AAAA---|                       |
  3.  |---198.51.100.1---------->|                       |
  4.  |---2001:DB8:0:2::1------->|                       |
  5.  |                          |                       |
  6.  |                          |--UDP Request, IPv6-->X|<---------
  7.  |                          |--UDP Request, IPv6-->X|  first
  8.  |                          |--UDP Request, IPv6-->X|  request/
  9.  |                          |                       |  response
 10.  |                          |--UDP Request, IPv4--->|  success
 11.  |                          |<-UDP Response, IPv4---|<---------
```

Figure 3: Message flow using UDP

In a UDP-based application (Figure 3), that would be from the DNS request (line 1) through one or more UDP Requests (lines 6-10) until a UDP Response is seen (line 11).

When using other transports, the methodology will have to be specified in context; it should measure the same event.

2.3. Happy Eyeballs metrics

The measurements taken are the duration of the interval from the initial DNS request until the session is seen to have been established, as described in Section 2.2. We are interested in the shortest and longest durations (which will most likely be those that send one SYN and succeed and those that send a SYN to each possible address before succeeding in one of the attempts), and the pattern of attempts sent to different addresses. The pattern may be to simply send an attempt every <time interval>, or may be more complex; as a result, this is in part descriptive.

ALL measurement events on the sending and receiving of messages SHALL be observed at the "Alice" attachment point and time stamps SHOULD be applied upon reception of the last bit of the IP information field. Use of a alternate timing reference SHALL be noted.

2.3.1. Metric: Session Setup Interval

Name: Session Setup Interval

Description: The session setup interval MUST be the time beginning with the first DNS query sent (observed at Alice's attachment), and ending with successful transport connection establishment (as indicated in line 12 of Figure 2, and line 11 of Figure 3). This interval is defined as the session setup interval.

This test will be run several times, once for each possible combination of destination address (configured on Bob) and failure mode (configured on Router 1).

Methodology: In the LAN analyzer trace, note the times of the initial DNS request and the confirmation that the session is open as described in Section 2.2. If the session is not successfully opened, possibly due to Alice aborting the attempt, the Session Setup Interval is considered to be infinite.

Units: Session setup time is measured in milliseconds.

Measurement Point(s): The measurement point is at Alice's LAN interface, both sending and receiving, observed using a program such as tcpdump running on Alice or an external analyzer.

Timing: The measurement program or external analyzer MUST run for a duration sufficient to capture the entire message flow as described in Section 2.2. Measurement precision MUST be sufficient to maintain no more than 0.1 ms error over a 60 second interval. 1 ppm precision would suffice.

2.3.2. Metric: Maximum Session Setup Interval

Name: Maximum Session Setup Interval

Description: The maximum session setup interval is the longest period of time observed for the establishment of a session as described in Section 2.3.1.

Methodology: see Session Setup Interval.

Units: Session setup time is measured in milliseconds.

Measurement Point(s): see Session Setup Interval.

Timing: The measurement program or external analyzer MUST run for a duration sufficient to capture the entire message flow as described in Section 2.2. Measurement precision MUST be sufficient to maintain no more than 0.1 ms error over a 60 second interval. 1 ppm precision would suffice.

2.3.3. Metric: Minimum Session Setup Interval

Name: Minimum Session Setup Interval

Description: The minimum session setup interval is the shortest period of time observed for the establishment of a session.

Methodology: see Session Setup Interval.

Units: Session setup time is measured in milliseconds.

Measurement Point(s): see Session Setup Interval.

Timing: The measurement program or external analyzer MUST run for a duration sufficient to capture the entire message flow as described in Section 2.2. Measurement precision MUST be sufficient to maintain no more than 0.1 ms error over a 60 second interval. 1 ppm precision would suffice.

2.3.4. Descriptive Metric: Attempt pattern

Name: Attempt pattern

Description: The Attempt Pattern is a description of the observed pattern of attempts to establish the session. In simple cases, it may be something like "Initial TCP SYNs to a new address were observed every <so many> milliseconds"; in more complex cases, it might be something like "Initial TCP SYNs in IPv6 were observed every <so many> milliseconds, and other TCP SYNs using IPv4 were observed every <so many> milliseconds, but the two sequences were independent." It may also comment on retransmission patterns if observed.

Methodology: The traffic trace is analyzed to determine the pattern of initiation.

Units: milliseconds.

Measurement Point(s): The measurement point is at Alice’s LAN interface, observed using a program such as tcpdump running on Alice or an external analyzer.

Timing: The measurement program or external analyzer MUST run for a duration sufficient to capture the entire message flow as described in Section 2.2. Measurement precision MUST be sufficient to maintain no more than 0.1 ms error over a 60 second interval. 1 ppm precision would suffice.

3. IANA Considerations

This memo asks the IANA for no new parameters.

Note to RFC Editor: This section will have served its purpose if it correctly tells IANA that no new assignments or registries are required, or if those assignments or registries are created during the RFC publication process. From the author's perspective, it may therefore be removed upon publication as an RFC at the RFC Editor's discretion.

4. Security Considerations

This note doesn't address security-related issues.

5. Acknowledgements

This note was discussed with Dan Wing, Andrew Yourtchenko, and Fernando Gont. In the Benchmark Methodology Working Group, Al Morton, David Newman, Sarah Banks, and Tore Anderson made comments.

6. Change Log

-00: Initial version - November, 2010

-01: Rewritten per suggestions by Al Morton, David Newman, and Sarah Banks.

-02: Clean-up per working group comments.

-03: Updated per Al Morton's and Tore Anderson's comments.

7. References

7.1. Normative References

[I-D.ietf-v6ops-happy-eyeballs]
Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", draft-ietf-v6ops-happy-eyeballs-03 (work in progress), July 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[RFC5461] Gont, F., "TCP's Reaction to Soft Errors", RFC 5461, February 2009.

Author's Address

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com