MMUSIC Internet-Draft Intended status: Standards Track Expires: August 18, 2014

# Happy Eyeballs Extension for ICE draft-reddy-mmusic-ice-happy-eyeballs-06

### Abstract

This document provides guidelines on how to make Interactive Connectivity Establishment (ICE) conclude faster in IPv4/IPv6 dualstack scenarios where broken paths exist.

### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of <u>BCP 78</u> and <u>BCP 79</u>.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <a href="http://datatracker.ietf.org/drafts/current/">http://datatracker.ietf.org/drafts/current/</a>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>http://trustee.ietf.org/license-info</u>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

# Table of Contents

<u>1</u> .	Introduction	•	•	÷	÷	<u>2</u>
<u>2</u> .	Notational Conventions					<u>2</u>
<u>3</u> .	Improving ICE Dual-stack Fairness					<u>3</u>
<u>4</u> .	Compatibility					<u>3</u>
<u>5</u> .	Example Algorithm for Choosing the Local Preference .					<u>4</u>
<u>6</u> .	IANA Considerations					<u>5</u>
<u>7</u> .	Security Considerations					<u>6</u>
<u>8</u> .	Acknowledgements					<u>6</u>
<u>9</u> .	Implementation Status					<u>6</u>
<u>9</u>	<u>9.1</u> . HappyE-ICE-Test					<u>6</u>
<u>10</u> .	Normative References					7
App	pendix A. Examples					7
Aut	thors' Addresses					10

### **<u>1</u>**. Introduction

There is a need to introduce more fairness in the handling of connectivity checks for different IP address families in dual-stack IPv4/IPv6 ICE scenarios. Section 4.1.2.1 of ICE [RFC5245] points to [RFC3484] for prioritizing among the different IP families. [RFC3484] is obsoleted by [RFC6724] but following the recommendations from the updated RFC will lead to prioritization of IPv6 over IPv4 for the same candidate type. Due to this, connectivity checks for candidates of the same type (HOST, RFLX, RELAY) are sent such that an IP address family is completely depleted before checks on the other address family are started. This results in user noticeable setup delays if the path for the prioritized address family is broken.

To avoid such user noticeable delays when either IPv6 or IPv4 path is broken, this specification encourages intermingling the different address families when connectivity checks are conducted. Introducing IP address family fairness into ICE connectivity checks will lead to more sustained dual-stack IPv4/IPv6 deployment as users will no longer have an incentive to disable IPv6. The cost is a small penalty to the address type that otherwise would have been prioritized.

### **2**. Notational Conventions

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 [<u>RFC2119</u>].

This document uses terminology defined in [RFC5245].

### 3. Improving ICE Dual-stack Fairness

Candidates SHOULD be prioritized such that a long sequence of candidates belonging to the same address family will be intermingled with candidates from an alternate IP family. For example, promoting IPv4 candidates in the presence of many IPv6 candidates such that an IPv4 address candidate is always present after a small sequence of IPv6 candidates, i.e., reordering candidates such that both IPv6 and IPv4 candidates get a fair chance during the connectivity check phase. This makes ICE connectivity checks more responsive to broken path failures of an address family.

An ICE agent can choose an algorithm or a technique of its choice to ensure that the resulting check lists have a fair intermingled mix of IPv4 and IPv6 address families. Modifying the check list directly can lead to uncoordinated local and remote check lists that result in ICE taking longer to complete or in the worst case scenario fail. The best approach is to modify the formula for calculating the candidate priority value described in ICE [RFC5245] section 4.1.2.1.

## 4. Compatibility

ICE [RFC5245] section 4.1.2 states that the formula in section 4.1.2.1 SHOULD be used to calculate the candidate priority. The formula is as follows:

priority =  $(2^{24})^{*}(type preference) +$  $(2^8)^*(local preference) +$ (2^0)\*(256 - component ID)

ICE [RFC5245] section 4.1.2.2 has guidelines for how the type preference and local preference value should be chosen. Instead of having a static value for IPv4 and a static value for IPv6 type of addresses for the local preference, it is possible to choose this value dynamically in such a way that IPv4 and IPv6 address candidate priorities ends up intermingled within the same candidate type (HOST, RFLX, RELAY).

The local and remote agent can have different algorithms for choosing the local preference value without impacting the synchronization between the local and remote check list.

The check list is made up by candidate pairs. A candidate pair is two candidates paired up and given a candidate pair priority as described in [RFC5245] section 5.7.2. Using the pair priority formula:

pair priority =  $2^{32*MIN(G,D)} + 2^{MAX(G,D)} + (G>D?1:0)$ 

Where G is the candidate priority provided by the controlling agent and D the candidate priority provided by the controlled agent. This ensures that the local and remote check lists are coordinated.

Even if the two agents have different algorithms for choosing the candidate priority value to get an intermingled set of IPv4 and IPv6 candidates, the resulting checklist, that is a list sorted by the pair priority value, will be identical on the two agents.

The agent that has promoted IPv4 cautiously i.e. lower IPv4 candidate priority values compared to the other agent, will influence the check list the most due to (2^32\*MIN(G,D)) in the formula.

These recommendations are backward compatible with a standard ICE implementation. If the other agent have IPv4 candidates with higher priorities due to intermingling, the effect is canceled when the checklist is formed and the pair priority formula is used to calculate the pair priority.

### 5. Example Algorithm for Choosing the Local Preference

The value space for the local preference is from 0 to 65535 inclusive. This value space can be divided up in chunks for each IP address family.

An IPv6 and IPv4 start priority must be given. In this example IPv6 starts at 60000 and IPv4 at 59000. This leaves enough address space to further play with the values if pr interface priorities needs to be added. The highest value should be given to the address family that should be prioritized.

	IPv6	IPv4					
	Start	Start					
65535	60k	59k	58k	57k	56k	55k	Θ
+	+	+	+	+	+	+	 +
	IF	ν6   ΙΡ\	4   IP	v6   IP	v4   IP	v6	
	(1	L)   (1	L)   (2	2)   (	2)   (	3)	
+	+	+	+	+	+	+	 +
	<-	N->					

The local preference can be calculated by the given formula:

 $local_preference = N*2*(Cn/Cmax)$ 

Where N is the absolute value of IPv6\_start-IPv4\_start. This ensures a positive number even if IPv4 is the highest priority. Cn is the number of current candidates of a specific IP address type and

candidate type (HOST, SRFLX, RELAY). Cmax is the number of allowed consecutive candidates of the same IP address type.

Using the values N=abs(60000-59000) and Cmax = 2 yields the following sorted local candidate list:

(1)	HOST	IPv6	(1)	Priority:	2129289471
(2)	HOST	IPv6	(2)	Priority:	2129289470
(3)	HOST	IPv4	(1)	Priority:	2129033471
(4)	HOST	IPv4	(2)	Priority:	2129033470
(5)	HOST	IPv6	(1)	Priority:	2128777471
(6)	HOST	IPv6	(2)	Priority:	2128777470
(7)	HOST	IPv4	(1)	Priority:	2128521471
(8)	HOST	IPv4	(2)	Priority:	2128521470
(9)	HOST	IPv6	(1)	Priority:	2128265471
(10)	HOST	IPv6	(2)	Priority:	2128265470
(11)	SRFLX	IPv6	(1)	Priority:	1693081855
(12)	SRFLX	IPv6	(2)	Priority:	1693081854
(13)	SRFLX	IPv4	(1)	Priority:	1692825855
(14)	SRFLX	IPv4	(2)	Priority:	1692825854
(15)	RELAY	IPv6	(1)	Priority:	15360255
(16)	RELAY	IPv6	(2)	Priority:	15360254
(17)	RELAY	IPv4	(1)	Priority:	15104255
(18)	RELAY	IPv4	(2)	Priority:	15104254

The result is an even spread of IPv6 and IPv4 candidates among the different candidate types (HOST, SRFLX, RELAY). The local\_preference value is calculated separately for each candidate type.

The resulting checklist will depend on the priorities of the remote candidates. It is not possible to ensure an even spread of IPv4 and IPv6 addresses unless both the remote and local sides uses the simple recommendations in this draft. It is worth noting that there is a good chance it will some effect even if the remote side does not support this. It will not break interoperability with other ICE implementations.

[[Q1: Need to take a closer look at how the unfeezing happens and how this afffects the component id is the sorting above. --palmarti]] [[Q2: The implementations of the algorithm does not implement pruning of the pairs. So the checklist is shorter in real life than the example in the appendix. --palmarti]]

### 6. IANA Considerations

None.

### 7. Security Considerations

STUN connectivity check using MAC computed during key exchanged in the signaling channel provides message integrity and data origin authentication as described in section 2.5 of [RFC5245] apply to this use.

#### 8. Acknowledgements

Authors would like to thank Dan Wing, Ari Keranen, Bernard Aboba, Martin Thomson, Jonathan Lennox and Balint Menyhart for their comments and review.

### 9. Implementation Status

[Note to RFC Editor: Please remove this section and reference to [RFC6982] prior to publication.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [<u>RFC6982</u>], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

#### 9.1. HappyE-ICE-Test

Organization: Private Initiative (palerikm@gmail.com)

Description: A private initiative to create working code to show how the recommendations in this draft can be implemented. The code is publicly available at github.

Implementation: https://github.com/palerikm/HappyE-ICE-Test

Level of maturity: The code only implements the parts that cover this draft and not a full ICE implementation. There is work in progress to get this into a full implementation, unfortunately that source code is not at the current time available to the public. It is currently not implementing the pruning of the checklist pairs as described in <u>section 5.7.3</u> of the ICE RFC.

Coverage: Implement this draft.

Licensing: BSD

Implementation experience: Fiddly. Please not that the developer also is author of this draft. The implementation also helped writing parts of this draft.

Contact: Paal-Erik Martinsen <palmarti@gmail.com>.

### **<u>10</u>**. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>BCP 14</u>, <u>RFC 2119</u>, March 1997.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", <u>RFC 3484</u>, February 2003.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", <u>RFC 5245</u>, April 2010.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", <u>RFC 6724</u>, September 2012.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", <u>RFC 6982</u>, July 2013.

Appendix A. Examples

Internet-Draft

\*\*\*\*\*\*\*\*\*\* Local Candidates (sorted) \*\*\*\*\*\*\*\* (1) HOST IPv6 (1) Priority: 2129289471 (2) HOST IPv6 (2) Priority: 2129289470 (3) HOST IPv4 (1) Priority: 2129033471 (4) HOST IPv4 (2) Priority: 2129033470 (5) HOST IPv6 (1) Priority: 2128777471 (6) HOST IPv6 (2) Priority: 2128777470 (7) HOST IPv4 (1) Priority: 2128521471 (8) HOST IPv4 (2) Priority: 2128521470 (9) HOST IPv6 (1) Priority: 2128265471 (10) HOST IPv6 (2) Priority: 2128265470 (11) SRFLX IPv6 (1) Priority: 1693081855 (12) SRFLX IPv6 (2) Priority: 1693081854 (13) SRFLX IPv4 (1) Priority: 1692825855 (14) SRFLX IPv4 (2) Priority: 1692825854 (15) RELAY IPv6 (1) Priority: 15360255 (16) RELAY IPv6 (2) Priority: 15360254 (17) RELAY IPv4 (1) Priority: 15104255 (18) RELAY IPv4 (2) Priority: 15104254 \*\*\*\*\*\*\*\* Remote Candidates \*\*\*\*\*\*\*\* (1) HOST IPv6 (1) Priority: 2129289471 (2) HOST IPv6 (1) Priority: 2129289471 (3) HOST IPv6 (1) Priority: 2129289471 (4) HOST IPv6 (2) Priority: 2129289470 (5) HOST IPv6 (2) Priority: 2129289470 (6) HOST IPv6 (2) Priority: 2129289470 (7) HOST IPv4 (1) Priority: 2129033471 (8) HOST IPv4 (1) Priority: 2129033471 (9) HOST IPv4 (2) Priority: 2129033470 (10) HOST IPv4 (2) Priority: 2129033470 (11) IPv6 (1) Priority: 1693081855 (12) IPv6 (2) Priority: 1693081854 (13) IPv4 (1) Priority: 1692825855 (14) IPv4 (2) Priority: 1692825854 (15) RELAY IPv6 (1) Priority: 15360255 (16) RELAY IPv6 (2) Priority: 15360254 (17) RELAY IPv4 (1) Priority: 15104255 (18) RELAY IPv4 (2) Priority: 15104254

The pairs have not been pruned a described in  $\underline{\text{section } 5.7.3}$  of the ICE spec.

\*\*\*\*\*\*\*\*\* CheckList \*\*\*\*\*\*\*\*
0 HOST 6(1) 2129289471 HOST 6(1) 2129289471(9145228645920719358)
1 HOST 6(1) 2129289471 HOST 6(1) 2129289471(9145228645920719358)

2 I	HOST	6(1)	2129289471 H	HOST (	6(1)	2129289471(9145228645920719358)
3 I	HOST	6(2)	2129289470 H	HOST (	6(2)	2129289470(9145228641625752060)
4 I	HOST	6(2)	2129289470 H	HOST (	6(2)	2129289470(9145228641625752060)
5 I	HOST	6(2)	2129289470 H	HOST (	6(2)	2129289470(9145228641625752060)
6 I	HOST	4(1)	2129033471 H	HOST 4	4(1)	2129033471(9144129134292431358)
71	HOST	4(1)	2129033471 H	HOST 4	4(1)	2129033471(9144129134292431358)
8	HOST	4(2)	2129033470 H	HOST 4	4(2)	2129033470(9144129129997464060)
9	HOST	4(2)	2129033470	HOST 4	4(2)	2129033470(9144129129997464060)
10	HOST	6(1)	2128777471	HOST	(1)	2129289471(9143029622665167358)
11	HOST	6(1)	2128777471	HOST	6(1)	2129289471(9143029622665167358)
12	HOST	6(1)	2128777471	HOST	6(1)	2129289471(9143029622665167358)
13	HOST	6(2)	2128777470	HOST	6(2)	2129289470(9143029618370200060)
1/	ност	6(2)	2128777470	ност	6(2)	2120280470(0143020618370200060)
15	цост	6(2)	2120777470	ПОСТ	6(2)	2120280470(0142020618270200000)
16	ПОЗТ	4(1)	2120777470	ПОЗТ	0(2)	2120022471(0141020111026870258)
17	поэт	4(1)	) 2120521471	позт	4(1)	2129033471(9141930111030079336)
10	HOST	4(1)	) 2120521471	HOST	4(1)	2129033471(9141930111030879358)
18	HUST	4(2)	) 2128521470	HUST	4(2)	2129033470(9141930106741912060)
19	HUST	4(2)	) 2128521470	HUST	4(2)	2129033470(9141930106741912060)
20	HUST	6(1)	) 2128265471	HOST	6(1)	2129289471(9140830599409615358)
21	HOST	6(1)	) 2128265471	HOST	6(1)	2129289471(9140830599409615358)
22	HOST	6(1)	) 2128265471	HOST	6(1)	2129289471(9140830599409615358)
23	HOST	6(2)	) 2128265470	HOST	6(2)	2129289470(9140830595114648060)
24	HOST	6(2)	2128265470	HOST	6(2)	2129289470(9140830595114648060)
25	HOST	6(2)	2128265470	HOST	6(2)	2129289470(9140830595114648060)
26	HOST	6(1)	2129289471	SRFLX	6(1)	1693081855(7271731200934593023)
27	SRFLX	6(1)	1693081855	HOST	6(1)	2129289471(7271731200934593022)
28	SRFLX	6(1)	1693081855	HOST	6(1)	2129289471(7271731200934593022)
29	SRFLX	6(1)	1693081855	HOST	6(1)	2129289471(7271731200934593022)
30	HOST	6(1)	2128777471	SRFLX	6(1)	1693081855(7271731200933569023)
31	HOST	6(1)	2128265471	SRFLX	6(1)	1693081855(7271731200932545023)
32	SRFLX	6(1)	1693081855	SRFLX	6(1)	1693081855(7271731200062177790)
33	HOST	6(2)	2129289470	SRFLX	6(2)	1693081854(7271731196639625725)
34	SRFLX	6(2)	1693081854	HOST	6(2)	2129289470(7271731196639625724)
35	SRFLX	6(2)	1693081854	HOST	6(2)	2129289470(7271731196639625724)
36	SRFLX	6(2)	1693081854	HOST	6(2)	2129289470(7271731196639625724)
37	HOST	6(2)	2128777470	SRFLX	6(2)	1693081854(7271731196638601725)
38	HOST	6(2)	2128265470	SRFLX	6(2)	1693081854(7271731196637577725)
39	SRELX	6(2)	1693081854	SRELX	6(2)	1693081854(7271731195767210492)
40	HOST	4(1)	2129033471	SRFLX	4(1)	1692825855(7270631689306305023)
41	SRELX	(-)	1692825855	HOST	4(1)	2129033471(7270631689306305022)
/12	SPELX	//(1)	1602825855	ност	1(1)	2120000111(1210001000000000000000000000
12	HOST	/(1)	2128521/71	SPELX	$\frac{1}{1}(1)$	1602825855(7270631680305281023)
40	CDEL V	ー(エ) イ(エ)	1602925955		-(+)	1602825855 (7270621688422880700)
44 15		、サ(エ) オ(つ)	1092020000		+(±) 4(2)	1602825857(7270621695011227725)
40		4(Z)	1602025054/0	JALLY	4(Z)	120022470(7270621696011237724)
40	SKELX	4(Z)	1602025054		4(Z)	2120022470(7270621605011233//24)
47	SKELX	4(Z)	1092825854		4(Z)	<pre>2129033470(7270031085011337724)</pre>
48	HUSI	4(2)	2128521470	SKELX	4(2)	1092825854(7270031685010313725)
49	SRELX	. 4(2)	1692825854	SRELX	4(2)	1692825854(7270631684138922492)

50	HOST	6(1)	2129289471	RELAY	6(1)	15360255 (	(65971797141799423)
51	RELAY	6(1)	15360255	HOST	6(1)	2129289471	(65971797141799422)
52	RELAY	6(1)	15360255	HOST	6(1)	2129289471	(65971797141799422)
53	RELAY	6(1)	15360255	HOST	6(1)	2129289471	(65971797141799422)
54	HOST	6(1)	2128777471	RELAY	6(1)	15360255 (	(65971797140775423)
55	HOST	6(1)	2128265471	RELAY	6(1)	15360255 (	(65971797139751423)
56	SRFLX	6(1)	1693081855	RELAY	6(1)	15360255 (	(65971796269384191)
57	RELAY	6(1)	15360255	SRFLX	6(1)	1693081855	(65971796269384190)
58	RELAY	6(1)	15360255	RELAY	6(1)	15360255 (	(65971792913940990)
59	HOST	6(2)	2129289470	RELAY	6(2)	15360254 (	(65971792846832125)
60	RELAY	6(2)	15360254	HOST	6(2)	2129289470	(65971792846832124)
61	RELAY	6(2)	15360254	HOST	6(2)	2129289470	(65971792846832124)
62	RELAY	6(2)	15360254	HOST	6(2)	2129289470	(65971792846832124)
63	HOST	6(2)	2128777470	RELAY	6(2)	15360254 (	(65971792845808125)
64	HOST	6(2)	2128265470	RELAY	6(2)	15360254 (	(65971792844784125)
65	SRFLX	6(2)	1693081854	RELAY	6(2)	15360254 (	(65971791974416893)
66	RELAY	6(2)	15360254	SRFLX	6(2)	1693081854	(65971791974416892)
67	RELAY	6(2)	15360254	RELAY	6(2)	15360254 (	(65971788618973692)
68	HOST	4(1)	2129033471	RELAY	4(1)	15104255 (	(64872285513511423)
69	RELAY	4(1)	15104255	HOST	4(1)	2129033471	(64872285513511422)
70	RELAY	4(1)	15104255	HOST	4(1)	2129033471	(64872285513511422)
71	HOST	4(1)	2128521471	RELAY	4(1)	15104255 (	(64872285512487423)
72	SRFLX	4(1)	1692825855	RELAY	4(1)	15104255 (	(64872284641096191)
73	RELAY	4(1)	15104255	SRFLX	4(1)	1692825855	(64872284641096190)
74	RELAY	4(1)	15104255	RELAY	4(1)	15104255 (	(64872281285652990)
75	HOST	4(2)	2129033470	RELAY	4(2)	15104254 (	(64872281218544125)
76	RELAY	4(2)	15104254	HOST	4(2)	2129033470	(64872281218544124)
77	RELAY	4(2)	15104254	HOST	4(2)	2129033470	(64872281218544124)
78	HOST	4(2)	2128521470	RELAY	4(2)	15104254 (	(64872281217520125)
79	SRFLX	4(2)	1692825854	RELAY	4(2)	15104254 (	(64872280346128893)
80	RELAY	4(2)	15104254	SRFLX	4(2)	1692825854	(64872280346128892)
81	RELAY	4(2)	15104254	RELAY	4(2)	15104254 (	(64872276990685692)

Authors' Addresses

Tirumaleswar Reddy Cisco Systems, Inc. Cessna Business Park, Varthur Hobli Sarjapur Marathalli Outer Ring Road Bangalore, Karnataka 560103 India

Email: tireddy@cisco.com

Prashanth Patil Cisco Systems, Inc. Bangalore India

Email: praspati@cisco.com

Paal-Erik Martinsen Cisco Systems, Inc. Philip Pedersens Vei 22 Lysaker, Akershus 1325 Norway

Email: palmarti@cisco.com