TAPS
Internet-Draft
Intended status: Experimental
Expires: December 3, 2017

K-J. Grinnemo
A. Brunstrom
P. Hurtig
Karlstad University
N. Khademi
University of Oslo
Z. Bozakov
Dell EMC Research Europe

June 2017

Happy Eyeballs for Transport Selection
draft-grinnemo-taps-he-03

Abstract

Ideally, network applications should be able to select an appropriate transport solution from among available transport solutions. However, at present, there is no agreed-upon way to do this. In fact, there is not even an agreed-upon way for a source end host to determine if there is support for a particular transport along a network path. This draft addresses these issues, by proposing a Happy Eyeballs framework. The proposed Happy Eyeballs framework enables the selection of a transport solution that according to application requirements, pre-set policies, and estimated network conditions is the most appropriate one. Additionally, the proposed framework makes it possible for an application to find out whether a particular transport is supported along a network connection towards a specific destination or not.

Status of This Memo

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

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 http://datatracker.ietf.org/drafts/current/.

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 December 3, 2017.

Table of Contents

1. Definitions 2
2. Introduction 2
3. Problem Statement 3
4. The Happy Eyeballs Framework 4
5. Design and Implementation Considerations 5
5.1. Candidate List Generation 5
5.2. Caching 7
5.3. Concurrent Connection Attempts 7
6. Example Happy Eyeballs Scenario 7
7. IANA Considerations 8
8. Security Considerations 8
9. Acknowledgements 8
10. References 8
10.1. Normative References 9
10.2. Informative References 9
Authors' Addresses 9

1. Definitions

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 RFC 2119 [RFC2119].

2. Introduction

Information services on the Internet come in varying forms, such as

web browsing, email, and on-demand multimedia. The main motivation behind the design of next-generation computer and communications networks is to provide a universal and easy access to these various types of information services on a single multi-service Internet. This means that all forms of communications, e.g., video, voice, data

and control signaling, along with all types of services -- from plain text web pages to multimedia applications -- are bonded in a single-service platform through Internet technology. To enable the next-generation networks, the TAPS Working Group suggests a decoupling between the transport service provided to an application, and the transport stack providing this transport service: An application requests an appropriate transport service on the basis of its transport requirements, and the available transport stack that best meets these requirements is selected. In case the most preferred transport stack is not supported along the network path to the destination, or is not supported by the end host, a less-preferred transport stack is selected instead. As a way to realize the selection of transport stacks, this document suggests a generalization of the Happy Eyeballs (HE) mechanism proposed in Wing et al. [RFC6555] which addresses the selection of complete transport solutions, and which lends itself to arbitrary transport selection criterias. The proposed HE mechanism targets connection-oriented transport solutions, and connectionless transport solutions provided they offer some reasonable way to determine their successful use between endpoints.

The HE mechanism was introduced as a means to promote the use of dual network stacks. Dual-stack client applications should be encouraged to try setting up connections over IPv6 first, and fall back to using IPv4 if IPv6 connection attempts fail. However, serializing tests for IPv6 and IPv4 connectivity can result in large connection latencies. HE for IPv6 minimizes the cost in delay by parallelizing attempts over IPv6 and IPv4. HE has also been proposed as an efficient way to find out the optimal combination of IPv4/IPv6 and TCP/SCTP to use to connect to a server [I-D.wing-tsvwg-happy-eyeballs-sctp]. The HE framework suggested in this document could be seen as a natural continuation of this proposal.

3. Problem Statement

Currently, there is no agreed-upon way for a source end host to select an appropriate transport service for a given application. In fact, there is no common way for a source end-host to find out if a transport stack is supported along a network path between itself and a destination end host. As a consequence, it has become increasingly difficult to introduce new transport stacks, and several applications, including many web applications, run over TCP although there are other transport protocols that better meet the requirements of these applications.

4. The Happy Eyeballs Framework

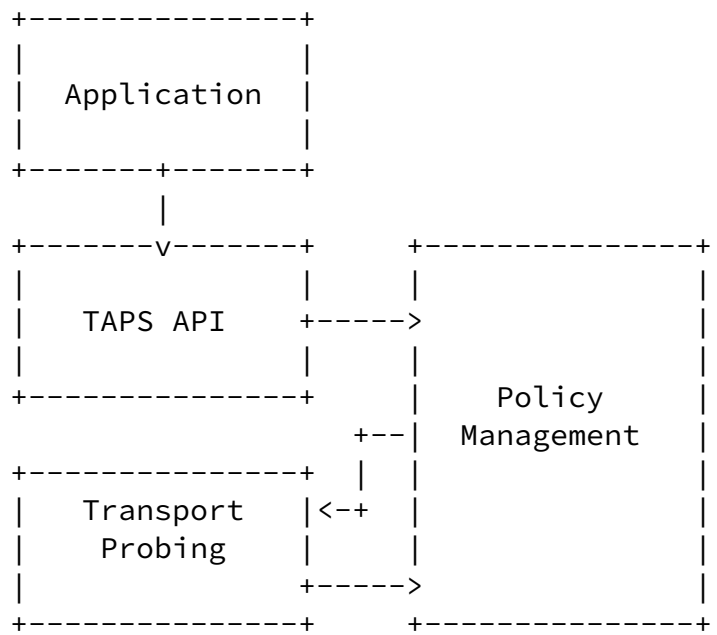


Figure 1: The Happy Eyeballs Framework.

The generalized HE mechanism proposed in this draft is carried out within the framework depicted in Figure 1. It comprises the following steps:

1. The Policy Management component takes as input application requirements from the TAPS API, stored information about previous

connection attempts (e.g., whether previous connection attempts succeeded or not), and network conditions and configurations. On the basis of this input and the policies configured in the system, the Policy Management component creates a list of candidate transport solutions, L, sorted in decreasing priority order. To be compliant with RFC 6555 [RFC6555], the Policy Management component SHOULD, in those cases there are no policies telling otherwise, following the host's address preference, something which usually means giving preference to IPv6 over IPv4.

2. It is the responsibility of the Transport Probing component to select the most appropriate transport solution. This is done by initiating connection attempts for each transport solution on L. To minimize the number of connection attempts that are initiated, the Transport Probing component SHOULD cache the outcome of connection attempts in a repository kept by the Policy Management component. The Policy Management component SHOULD in turn only include those transport solutions on L that have not been previously attempted, have valid successful connection-attempt

cache entries, or have previously been attempted but whose cached connection-attempt entries have expired. Cached connection-attempt results SHOULD be valid for a configurable amount of time after which they SHOULD expire and have to be repeated. The transport solutions on L are initiated in priority order. The difference in priority between two consecutive candidates, C1 and C2, is translated according to some criteria to a delay, D. D then governs the delay between the initiation of the connection attempts C1 and C2.

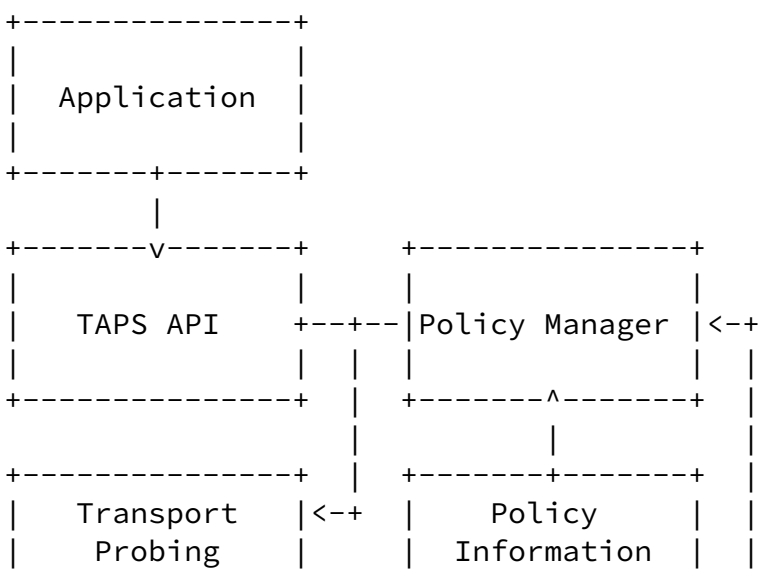
3. After the initiation of the connection attempts, the Transport Probing component waits for the first or winning connection to be established, which becomes the selected transport solution. For the Transport Probing component to be able to efficiently use the connection-attempt cache, already-initiated, non-winning connection attempts SHOULD be given a fair chance to complete. In that way, the connection-attempt cache will be provided with a fairly accurate knowledge of which transport solutions work and does not work against frequently visited transport endpoints. Moreover, it MAY be beneficial to let those transport solutions which have a higher priority than the winning transport solution, live a predetermined amount of time after their establishment, since this enables the reuse of already established connections

in later application requests.

5. Design and Implementation Considerations

This section discusses implementation issues that should be considered when a HE mechanism is designed and implemented on the basis of the HE framework proposed in this document.

5.1. Candidate List Generation



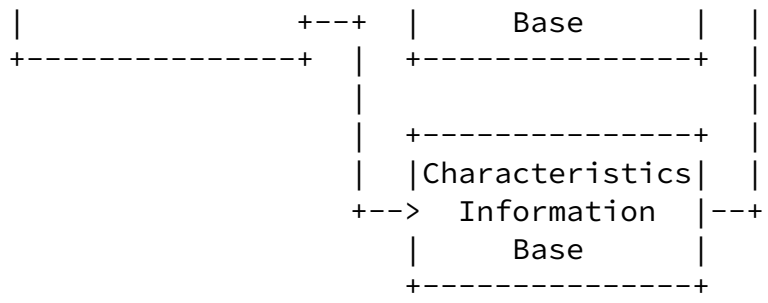


Figure 2: Principle Design of the NEAT Happy Eyeballs Framework.

There are several ways in which the list of candidate transport solutions, L, could be created by the Policy Management component. For example, L could be a list of all available transport solutions in an order that, except for following the host's address preference, is arbitrary; another, more sophisticated, way of creating the list of candidate transport solutions is the one employed by the NEAT System.

The NEAT System is developed as part of the EU Horizon 2020 project, "A New, Evolutive API and Transport-Layer Architecture for the Internet" (NEAT) [NEAT-Webb], and aims to provide a flexible and evolvable transport system that aligns with the charter of the TAPS Working Group. In the NEAT System [NEAT-Git], the HE framework is realized as shown in Figure 2. As follows, the Policy Management component comprises three components in the NEAT HE framework: a Policy Manager (PM), a Policy Information Base (PIB), and a Characteristics Information Base (CIB). PIB is a repository that stores a collection of policies that map application requests to transport solutions, i.e., map application requests to appropriately configured transport protocols, and CIB is a repository that stores information about previous connection attempts, available network

interfaces, supported transport protocols etc. The PM takes as input application requirements from the TAPS API, and information from PIB and CIB. On the basis of this input, the PM creates L.

5.2. Caching

As pointed out in RFC 6555 [RFC6555], a HE algorithm should not waste networking resources by routinely making simultaneous connection

attempts. To this end, the HE algorithm should cache the outcome of previous connection attempts to the same peer. The cache lifetime is considered system dependent and should be set on a case-by-case basis. The impact and efficiency of the HE algorithm have been evaluated in [Papastergiou16]. The paper suggests that caching significantly reduces the CPU load imposed by a HE mechanism. It also indicates that the internal-memory footprint of a HE mechanism is essentially the same as for single-flow establishments.

5.3. Concurrent Connection Attempts

As mentioned in Section 4, it is the responsibility of the Transport Probing component to choose the most appropriate transport solution on the list of candidate transport solutions, L. Often this implies that several transport solutions need to be tried out, something which should not be carried out sequentially, but concurrently or partly overlapping depending on the transport-solution priorities. The way this is done is implementation dependent and varies between platforms. The NEAT library [NEAT-Git], which implements the HE framework herein, is built around the libuv asynchronous I/O library [LIBUV] and uses an event-based concurrency model to realize the concurrent initialization of connection attempts. The rationale behind using an event-based concurrency model is at least twofold: The first is that correctly managing concurrency in multi-threaded applications can be challenging with, for example, missing locks or deadlocks. The second is that multi-threading typically offers little or no control over what is scheduled at a given momemt in time. Given the complexity of building a general-purpose scheduler that works well in all cases, sometimes the OS will schedule work in a manner that is less than optimal. Those in favor of threads argue that threads are a natural extension of sequential programming in that it maps work to be executed with individual threads. Threads are also a well-known and understood parts of OSes, and are mandatory for exploiting true CPU concurrency.

6. Example Happy Eyeballs Scenario

Consider a scenario in which an IPv6-enabled client using the NEAT System wishes to setup a connection to a server. Assume both the client and server support SCTP and TCP. The Policy Management is

queried about feasible transport solutions to connect to the server.

In the NEAT System, this results in PM retrieving information about network connections against this server from the CIB, e.g., supported transport protocols and the outcome of previous connection attempts. In our scenario, the PM learns from the CIB that the server supports SCTP and TCP, and, for the sake of this example, let us assume that the PM is also informed that previous connection attempts against this server, using both SCTP and TCP, were successful. Next, the PM retrieves applicable policies from the PIB, and combines these policies with the previously retrieved CIB information. We assume in this example that the SCTP transport solution has a higher priority than the TCP solution. As a next step, the PM puts together the feasible candidate transport solutions in a list with SCTP over IPv6 placed at the head of the list followed by TCP over IPv6, and supplies this list to the Transport Probing component. The Transport Probing component traverses the candidate list, and initiates a connection attempt with SCTP against the server followed after a short while (governed by the difference in priorities between the SCTP and TCP transport solutions) by a connection attempt with TCP against the server. In our example, assume both connection attempts are successful, however, the SCTP connection attempt completes before the TCP attempt. The Transport Probing component caches in the CIB the SCTP connection attempt as successful, and returns the SCTP connection as the winning connection. When the TCP connection is established some time later, the Transport Probing component caches that connection attempt as successful as well.

7. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

8. Security Considerations

Security will be considered in future versions of this document.

9. Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644334 (NEAT). The views expressed are solely those of the author(s).

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <http://www.rfc-editor.org/info/rfc6555>.

10.2. Informative References

[I-D.wing-tsvwg-happy-eyeballs-sctp]
Wing, D. and P. Natarajan, "Happy Eyeballs: Trending Towards Success with SCTP", draft-wing-tsvwg-happy-eyeballs-sctp-02 (work in progress), October 2010.

[LIBUV] libuv -- Asynchronous I/O Made Simple, "http://libuv.org", March 2017.

[NEAT-Git]
A New, Evolutive API and Transport-Layer Architecture for the Internet (NEAT), "https://github.com/NEAT-project/neat", March 2017.

[NEAT-Webb]
NEAT -- A New, Evolutive API and Transport-Layer Architecture for the Internet, "https://www.neat-project.org", March 2017.

[Papastergiou16]
Papastergiou, G., Grinnemo, K-J., Brunstrom, A., Ros, D., Tuexen, M., Khademi, N., and P. Hurtig, "On the Cost of Using Happy Eyeballs for Transport Protocol Selection", July 2016.

Authors' Addresses

Karl-Johan Grinnemo
Karlstad University
Universitetsgatan 2
Karlstad 651 88
Sweden

Phone: +46 54 700 24 40

Email: karl-johan.grinnemo@kau.se

Anna Brunstrom
Karlstad University
Universitetsgatan 2
Karlstad 651 88
Sweden

Phone: +46 54 700 17 95
Email: anna.brunstrom@kau.se

Per Hurtig
Karlstad University
Universitetsgatan 2
Karlstad 651 88
Sweden

Phone: +46 54 700 23 35
Email: per.hurtig@kau.se

Naeem Khademi
University of Oslo
PO Box 1080 Blindern
Oslo N-0316
Norway

Email: naeemk@ifi.uio.no

Zdravko Bozakov
Dell EMC Research Europe
Ovens, Co.
Cork
Ireland

Phone: +353 21 4945733
Email: Zdravko.Bozakov@dell.com