Happy Eyeballs Extension for Multiple Interfaces

draft-chen-mif-happy-eyeballs-extension-01

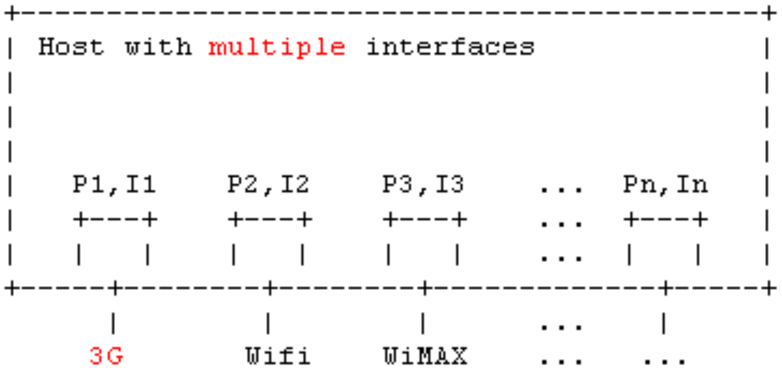
Gang Chen (chengang@chinamobile.com)

Carl Williams(carlw@mcsr-labs.org)

28-March-2011

Happy Eyeballs Extension Overview

- Happy Eyeballs has described how a dual-stack client can determine the functioning path to a dual-stack server
- For now, the assumption here is single-homed host
- We will propose to extend happy eyeballs algorithm to fit into multiple interfaces environment.
- To be specific



Each interface will be configured with weighting coefficient, which is composed of pair values

- Value **I** is defined to indicate preference of interfaces selection

- Value **P** is a indication to choose proper IP family for individual interface

Objective

A client with multiple interface could determine the optimal interface and appropriate IP address family for this interface

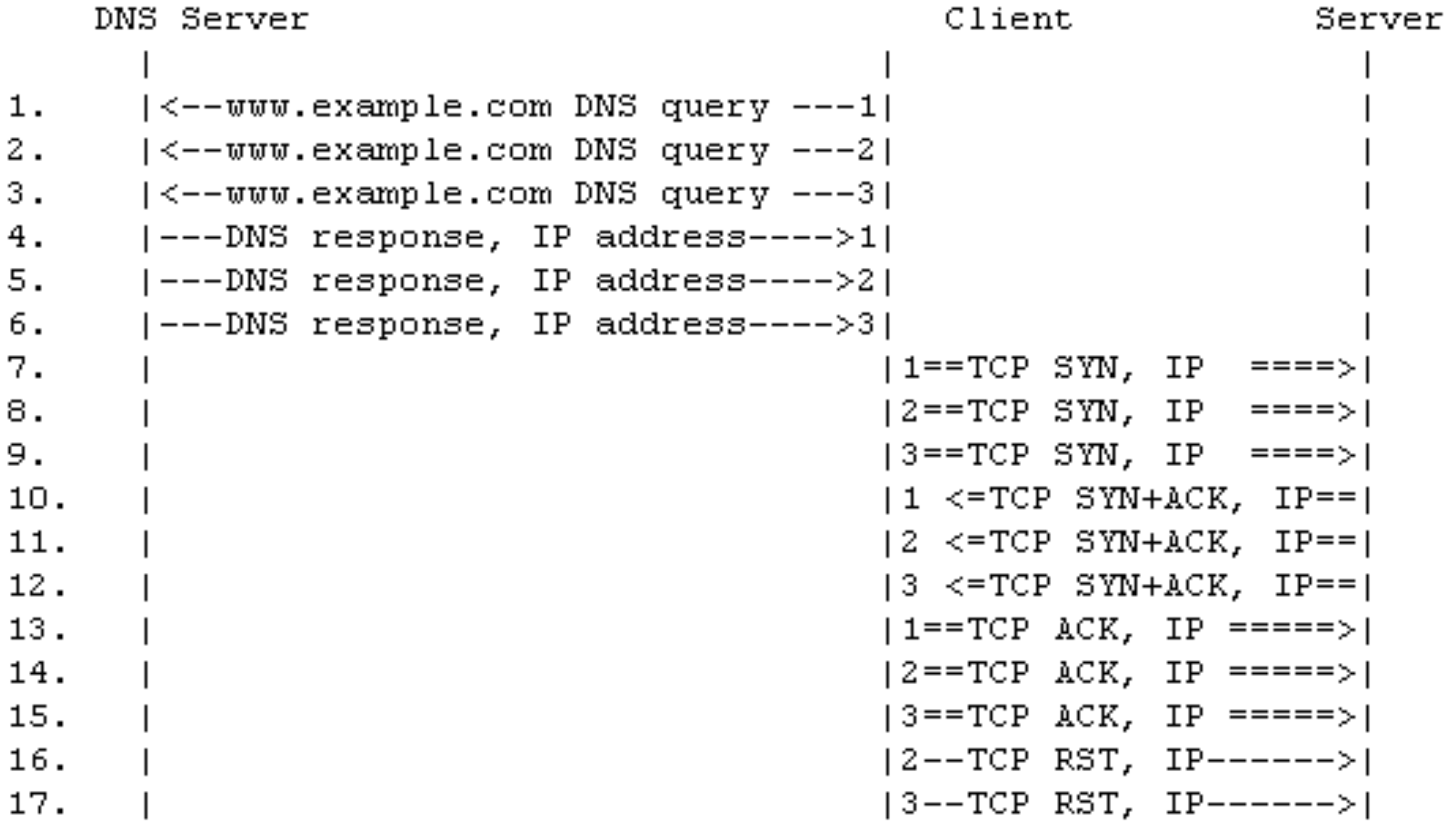
Algorithm for Interface Selection

- Each interface is configured with one value, **I**. **I** is served as an indication to identify which interface is preferred for a specific destination or hostname
- A positive value indicates preference of specific interface compared to others
- When one interface defeats others, the corresponding value **I** will be set to positive value. Other interfaces will be set negative value orderly according to caused time for TCP connection, like -1, -2, etc..
- The selection of a particular interface from the viable set implies a selection of one particular network path in preference to other viable paths

IPv4/IPv6 Selection Algorithm for Individual Interface

- For a specific interface in a dual-stack single interface node, the choice of IP address family relies on Happy Eyeballs algorithm

Example of Extended Algorithm



```
    DNS Server                             Client           Server
      |                                 |                     |
1.    |<--www.example.com DNS query ---1|                     |
2.    |<--www.example.com DNS query ---2|                     |
3.    |<--www.example.com DNS query ---3|                     |
4.    |---DNS response, IP address---->1|                     |
5.    |---DNS response, IP address---->2|                     |
6.    |---DNS response, IP address---->3|                     |
7.    |                                 |1==TCP SYN, IP  ====>|
8.    |                                 |2==TCP SYN, IP  ====>|
9.    |                                 |3==TCP SYN, IP  ====>|
10.   |                                 |1 <=TCP SYN+ACK, IP==|
11.   |                                 |2 <=TCP SYN+ACK, IP==|
12.   |                                 |3 <=TCP SYN+ACK, IP==|
13.   |                                 |1==TCP ACK, IP =====>|
14.   |                                 |2==TCP ACK, IP =====>|
15.   |                                 |3==TCP ACK, IP =====>|
16.   |                                 |2--TCP RST, IP------>|
17.   |                                 |3--TCP RST, IP------>|
```

Interface 1 competes with another two and firstly finish TCP connection. So value **I** will be set to 1. And second one is set to -1; the last one is set to -2. The interface 2 and 3 will then send RST to remote IP peer for release TCP sessions.

Additional Considerations

- Usage Scope
  - Happy Eyeballs is targeting to HTTP context, but it is useful and applicable to other time-sensitive applications
- Flow Continuity
  - Interface changing happens at the beginning of new session. So, there is no flow continuity issues for ongoing TCP session. Dynamic movement of traffic flows are addressed by other IETF protocols as well
- Default Address Selection
  - If more than one IPv6 address is assigned to the interface, the native IPv6 address is given preference
- DNS server Selection
  - The extended Eyeballs algorithm for described herein is independent of DNS server selection routings implemented on the multi-interface client

Some comments from mailing-list

- Is the value of I initially zero for any destination/ hostname that has not yet been connected to?
  - The value of I is supposed to be zero initially
- What about incrementing I for a destination address that matches a more specific route? Hence initial connection attempt would be sent over the interface that has matching route and other interfaces would be tried only if no reply on the preferred one?
  - We could take routing information into account for computing value I
  - Preliminary enhancement is to set I as positive value for optimal interface which is aligned with routing information
  - Other interfaces send connections attempts simultaneously to set I as negative value orderly

Some comments from mailing-list

- How does the value of I relate to time? (i.e. if I is -1 does the node wait 10ms before trying corresponding less-preferred interface)
  - How about absolute value of I*10 milliseconds which in line with original Happy Eyeballs proposal
- Would the "I" be nonzero for those interfaces over which DNS suffix matching the requested name has been received on?
  - The extended Eyeballs algorithm is proposed to be independent with DNS server selection
  - Multiple rules might make algorithm complicated and cause potential confliction

Questions & Comments