hiprg
Internet-Draft
Intended status: Informational
Expires: March 4, 2011

Jun. Wang, Ed.
Jiong. Shen
ZTE Corporation

August 31, 2010

HIP Service Overlay Study
draft-wang-hiprg-service-overlay-01.txt

Abstract

This draft is a HIP service overlay study document, it presents several disadvantages of current HIP protocol and then takes a brief introduction of two existing alternative solutions. Finally, the authors propose a HIP service overlay architecture.

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 March 4, 2011.

Table of Contents

1. Introduction

HIP protocol employed an end-to-end communication model, which makes the deployment very simple - it's not necessary to deploy additional expensive infrastructure. But this kind of design incurs some weaknesses mentioned as following.

#1. Before one host can communicate with another host, it MUST initiate a HIP 4-way handshake, and then initiate a TCP handshake and other transport or application connections. If the two hosts have big RTT, it leads to a long connection delay and downgrades the user experience.

#2. When all the underlying links of one host get broken, the initiator doesn't know the link had broken until it has gotten O(n^2) fail attempts for each host it connected, in which the n is the number of one multihoming host's locators.

#3. Existing widely deployed telecomm network employs pre-shared key security mechanism rather than PKI. So if HIP can support pre-shared key authentication, the existing infrastructure can be reused.

#4. Since HIP mobility mechanism uses end to end connection, if a HIP host's IP address changed, it must sends an update message to its connected peer. Such design makes the mobility possible even if infrastructure does not involved, but it also causes two weaknesses: 1)If the connected two hosts have big RTT or if the HIP host has too many connections, the update may be time-consuming and leads to very high handover delay. 2)If two hosts of one connection change their IP addresses simultaneously, the update could never been successful only after using some anchor mechanism such as RVS, but as RVS only forwarding HIP signaling message, some data packages may missing.

1.1. Requirements Language

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. Terminology

HAP: Host identity protocol(HIP) access peer, the attachment point for HIP host provided by the HIP service overlay.

HSN: HIP service overlay super node, the actual HIP service node. They provide the rendezvous services and route the HIP packets originally from any authorized HIP host. At most cases, the HAPs are

also HSNs.

3. Alternative solutions

3.1. i3

i3(Internet Indirection Infrastructure) offers a rendezvous-based communication abstraction; applications can easily implement a variety of communication services, such as multicast, anycast, and mobility, on top of this communication abstraction. Sources send packets to a logical identifier, and receivers express interest in packets sent to an identifier. Delivery is best-efforts like in today's Internet, with no guarantees about packet delivery and packets are not stored in i3, they are only forwarded.

i3 is an overlay network which consists of a set of servers that store triggers(the identifier and the address binding) and forward packets between i3 nodes and to end-hosts. Identifiers and triggers have meaning only in this i3 overlay. i3 uses Chord lookup protocol, also in principle, i3 can use any of the proposed P2P lookup systems such as CAN, Pastry and Tapestry etc.

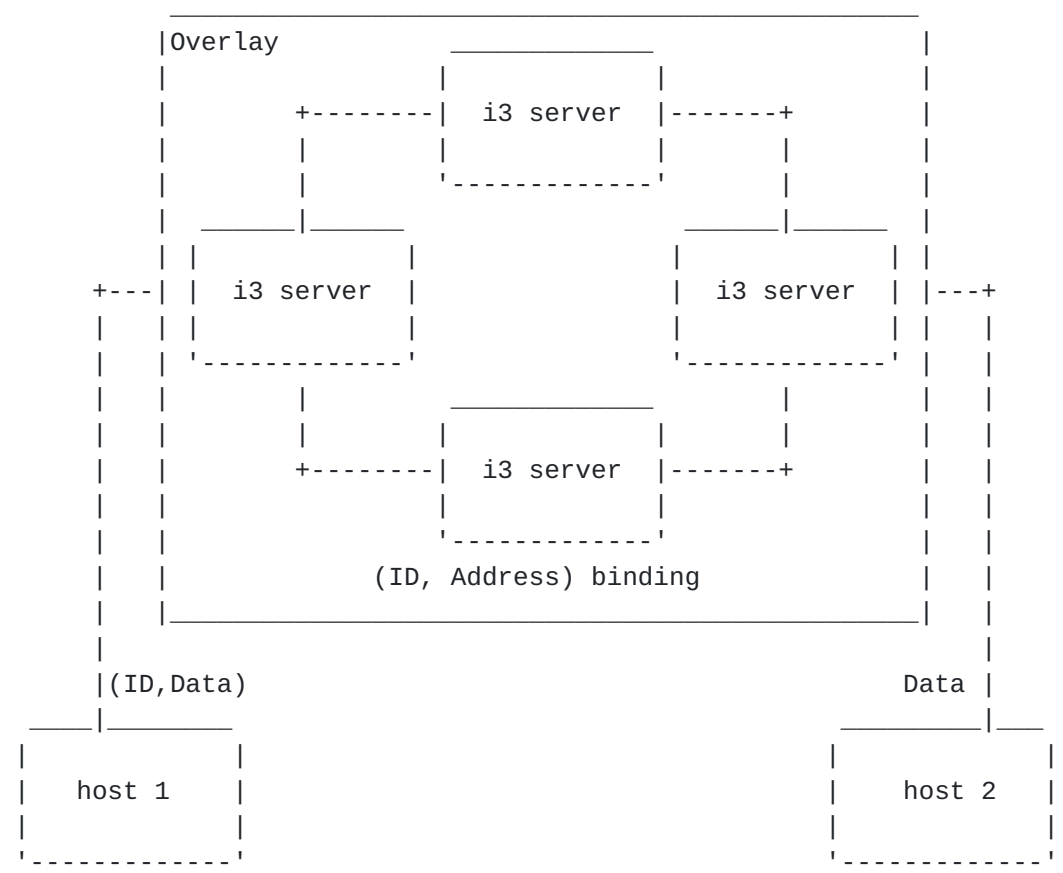

Figure 1: i3 Architecture

i3 uses i3 header to carry the identifier information on top of IP. But as i3 is not a layer 3.5 protocol, the i3 header is implemented on top of TCP or UDP. (Source from i3 implementation of Berkeley University. http://i3.cs.berkeley.edu/). For the four issues listed in introduction part, as i3 does not need HIP handshake, the issue #1 goes away; and as hosts always send packages to i3 infrastructure, and update i3 infrastructure about it's IP address changes, so issue #2, #3 and #4 also can be resolved. But as i3 is not a 3.5 layer protocol, it's not transparent support layer four and uplayer applications, and difficult to standardization.

3.2. Hi3

Hi3(Host Identity Indirection Infrastructure) recommends using the Internet Indirection Infrastructure(i3) to relay Host Identity Protocol(HIP) handshake packets, thereby serving as a control plane. Using i3 as a control plane for HIP in Hi3 improves protection from DoS attacks and provides an initial rendezvous service. Hi3 still uses end to end HIP connection as data plane after HIP handshake.

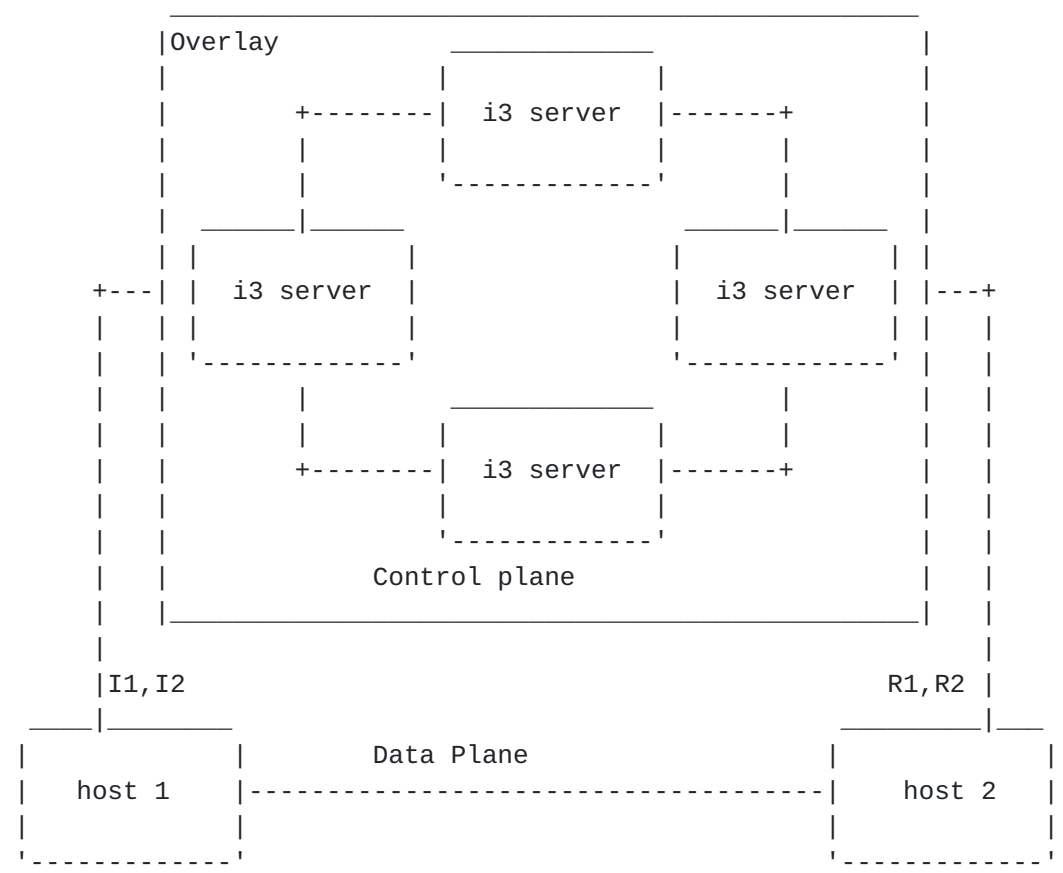

Figure 2: Hi3 Architecture

Hi3 uses i3 as control plane for forwarding HIP handshake packages I1, I2, R1 and R2, but still uses end to end PKI infrastructure and end to end HIP connection as data plane, so the issues #1, #2, #3 and #4(1) listed in introduction part still exist.

4. HIP service overlay

4.1. Architecture

The HIP service overlay provides HIP service to the HIP hosts, it also acts as a rendezvous server. If the hosts want to access the HIP service provided by the HIP service overlay, they MUST register in the HIP service overlay. HIP service overlay stores the registration information including locator and identifier bindings according to its own distributed algorithm.

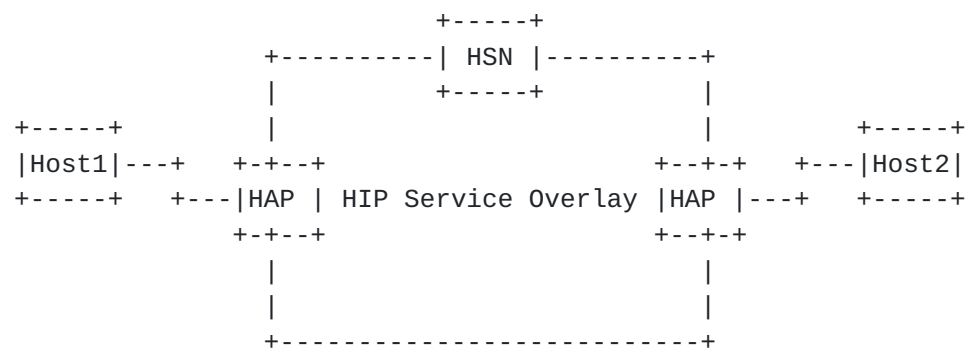

Figure 3: HIP Service Overlay Architecture

After the host registered to the HIP service overlay successfully, it established HIP connections with one or more HAPs, then the host can choose arbitrary one HAP to initiate a communication to another host, no further HIP handshakes are required.

The HAPs and HSNs comprise a HIP service overlay, where the HIP identifier/locator bindings are stored according to the DHT algorithm. HIP identifiers are treated as DHT resource keys and hashed into a fixed length bit sequence to formulate resource IDs. In most cases, the classical DHT algorithms with the O(LogN) search complexity have poor performance, we must choose some other algorithms with better performance. The one hop DHT can meet the rigid performance requirement, but it has the scalability issues. We'll take further discussion in later section.

4.2. Registration procedure

When the HIP host wants to register to a HIP service overlay, it can inquire DNS for the HIP proxy(HAP) addresse or by inquire some static configured server. The server may reture several HAP addresses, which are expected to be used in upcoming connection. HIP host then formulates a HIP I1 packet as a normal registration request. Then the HAP informs the HIP host expected authentication method and nonce in R1 packet. If the HIP host succeeds in authentication, HAP returns a positive response in R2 packet. And the HIP connection is established between host and HAP.

4.3. Initiating a communication

When a HIP host initiates a connection to a new host, e.g. sends a TCP SYN packet in the HIP data packet to a new host, the host's HIP stack inserts a proxy address, which is gotten from registration procedure, into the HIP packet and routing the packet to the HAP which has the proxy address. After the HIP service overlay received

an initial HIP packet, it searches whether the destination identifier has been attached, if found, HIP service overlay directs the packet to the HAP where destination host attached, then the HAP forwards packet to the final destination.

In this specification, the HIP host does not need to initiate a HIP 4-way handshake before establishing a transport connection, thus reducing the extra connection delay.

4.4. Updating location

When the location of a HIP host is changed, the HIP host MUST notify the HAP it attached for this change by HIP update, and then all the packages to this host will be forwarded to this new location.

In this specification, the HIP host does not need to notify the location change to the communicating hosts directly, thus reducing the extra connection delay.

4.5. Scalability and performance

Theoretically, the HIP service overlay can be deployed in any static or dynamic topology. For example, we can adopt a hierarchical topology, where the routing path is decided by the HIP prefix just like the PSTN network, but end to end connections in hierarchical network often traverse too many intermediate nodes, as a result, the connection gets high delay. So we recommend more flat topology, such as some advanced DHT networks. One hop DHT has the best performance, but it only supports very limited number of nodes, shouldn't be used to build a global HIP service overlay. The constant hops DHT algorithms trade off between the scalability and performance, they can accommodate enough number of hosts and have relatively low search delay, Kelips is an eligible algorithm.

Another scalability issue is caused by the truth that the HIP service overlay forwards all of the traffic either the control plane or the data plane. Despite one $1000 pc server can handle several Gbps IP traffic, but the bandwidth consumption of client PCs or other terminal equipments increases fast than before, so all the traffic pass through the HIP service overlay may be costly. We suggest that only the selected services can be deployed on the top of HIP service overlay, the HIP host can download the valid HIP prefix of the HIP service overlay, then if the protocol stack receive a socket send request, it can check if the destination address match an existing HIP service overlay prefix and sends the packet to a proper next hop.

We are doing prototype for the performance measure.

5. Acknowledgements

This HIP service overlay is based on ideas coming from conversations and discussions with a number of people in the HIP and HIPRG communities.

6. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see the update of RFC 2434 [I-D.narten-iana-considerations-rfc2434bis] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

7. Security Considerations

TBD.

8. References

8.1. Normative References

[Hi3] Nikander, P., Arkko, J., and B. Ohlman, "Host Identity Indirection Infrastructure (Hi3)", 2004.

[KELIPS] Gupta, Indranil., Birman, Ken., Linga, Prakash., Demers, Al., and Robbert. Renesse, "Building an Efficient and Stable P2P DHT Through Increased Memory and Background Overhead", 2003.

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

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[i3] Stoica, Ion., Adkins, Daniel., Zhuang, Shelley., and Scott. Shenker, "Internet Indirection Infrastructure", 2002.

8.2. Informative References

[I-D.narten-iana-considerations-rfc2434bis]
Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", draft-narten-iana-considerations-rfc2434bis-09 (work in progress), March 2008.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

Appendix A. Additional Stuff

Authors' Addresses

Jun Wang (editor)
ZTE Corporation
Nanjing, 210012
China

Phone: +86 25 52877648
Email: wang.jun17@zte.com.cn

Jiong Shen
ZTE Corporation
Nanjing, 210012
China

Phone: +86 25 52877648
Email: shen.jiong@zte.com.cn