HIP Working Group
Internet Draft
Intended status: Informational
Expires: September 2009

Gyu Myoung Lee
Jun Kyun Choi
ICU
Seng Kyoun Jo
Jeong Yun Kim
ETRI

March 9, 2009

HIP Extensions for Object to Object Communications
draft-lee-hip-object-02.txt

Status of this Memo

Abstract

This document explains the concept of object to object communications and specifies naming and addressing issues for object identification. In order to use Host Identity Protocol (HIP) for object to object communications, this document provides the extended architecture of HIP according to mapping relationships between host and object(s). In addition, packet formats and considerations for HIP extensions concerning object are specified.

Conventions used in this document

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.

Table of Contents

1. Introduction

The role of Host Identity Protocol (HIP) is the separation between the location and identity information by introducing a new cryptographic name space which is called Host Identity (HI). It provides enhanced network security as well as easy management of mobility and multi-homing [RFC4423].

The one of new capabilities for future network will be the ubiquitous networking such as the Internet of things. This networking capability requires "Any Services, Any Time, Any Where and Any Devices" operation. In order to connect objects (e.g., devices and/or machines) to large databases and networks, a simple, unobtrusive and cost-effective system of item identification is crucial. The concept of host should be extended to support all of objects. However, there is no consideration for new type of objects (e.g., contents, RFID tags, sensors, etc) as end points.

This document explains object to object communications. For identification of network entities, we consider new type of identifiers (e.g., RFID code, content ID, etc) for object and describe specific requirements for object identification in naming and addressing point of view.

In order to use HIP for object to object communication, this document provides the extended architecture of HIP according to mapping relationship between host and object(s). In addition, packet formats and considerations for HIP extensions are specified.

2. Object to Object Communications

For ubiquitous networking [Y.NGN-UbiNet], future network will require the extensions of networking functionalities to all objects. New networking concept will be considered for networking capabilities to support various classes of applications/services which require "Any Services, Any Time, Any Where and Any Devices" operation using Internet. This networking capability should support human-to-human, human-to-object (e.g., device and/or machine) and object-to-object communications.

There are many different kinds of devices connecting to the network supported for ubiquitous networking in Internet. RFID tag, sensors, smart cards, medical devices, navigation devices, vehicles as well as

the existing personal devices such as PC, Personal Digital Assistant (PDA), etc., are examples of these. This document considers that the end points which are not always humans but may be objects such as devices /machines, and then expanding to small objects and parts of objects.

Thus, object to object communications will be provided using the new concept of end points considering object. This document focuses on how to support object to object communications using extensions of existing HIP.

3. Object Identification

3.1. Classification of network entities to be identified

There are several network entities to be identified in the network. These network entities have a layered architecture and are used for naming, addressing and routing.

o Services (i.e., information related to applications/services)

o End points (i.e., global unique identifier)

o Location (i.e., IP address)

o Path (i.e., routing)

In particular, for object to object communications, information for several kinds of object on top of end points should be identified in the network.

3.2. Identification codes

Identification of all objects for providing end-to-end connectivity in ubiquitous networking environment is crucial. Identifier is capable of identifying all objects and facilitates objects-to-objects communications. In particular, the globally unique identifier enables a lot of applications including item tracking, access control, and protection, etc [1].

There are many kinds of identifiers such as E.164 number code, Extended Unique Identifier (EUI)-64, Media Access Control (MAC) address, Uniform Resource Identifier (URI)/ Uniform Resource Locator (URL), etc.

These identification codes can be classified as follows.

- Service IDs: include RFID, Content ID, telephone number, URL/URI, etc

- Communication IDs: include session/protocol ID, IP address, MAC address, etc

3.3. Examples of service IDs for objects

3.3.1. RFID

The identification codes, so-called Electronic Product Code (EPC), for RFID/sensors are very important in ubiquitous networking environment. An EPC is simply a number assigned to an RFID tag representative of an actual electronic product code. Their value is that they have been carefully characterized and categorized to embed certain meanings within their structure. Each number is encoded with a header, identifying the particular EPC version used for coding the entire EPC number. An EPC manager number is defined, allowing individual companies or organizations to be uniquely identified; an object class number is present, identifying objects used within this organization, such as product types. Finally, a serial number is characterized, allowing the unique identification of each individual object tagged by the organization [2].

3.3.2. Content ID

The Content ID is a unique identifier that can specify and distinguish any kind of digital content that is distributed. As a unique code attached to a content object, the Content ID serves well enough as an identifier, but actually it is much more than just that. It is also the key to a complete set of attribute information about a content object stored as metadata including the nature of the contents, rights-related information, information about distribution, and more. The Content ID provides the key enabling metadata to be uniquely associated with a particular digital object [3].

3.4. Requirements for naming and addressing using object identification

The layered architecture of naming and addressing requires specific processing capabilities at each layer. Each user/object in service layer identifies by identity like name with a set of attributes of an entity. An attribute can be thought of as metadata that belongs to a specific entity in a specific context, some of which could to be highly private or sensitive. The identity should be associated with

service IDs (RFID, content ID, telephone number, URI/URL, etc) through identification and authorization.

As shown in Figure 1, each service ID should be associated with communication IDs (session/protocol ID, IP address, MAC address, etc) through mapping/binding [Y.ipv6-ID].

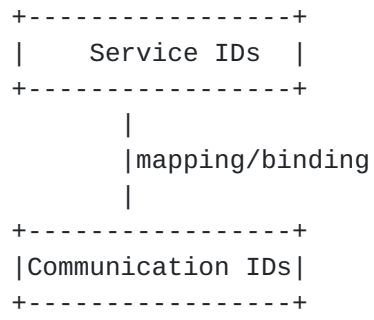


Figure 1 Mapping/binding for naming and addressing

An ID resolution server such as Domain Name System (DNS), can provide a function to translate the identifier of object into service /communication ID to access networking services provided by database/application servers.

How to map/bind IP address (i.e., communications IDs) with other identifiers (i.e., service IDs) for providing end-to-end IP connectivity is challenging issue.

Additionally, the following features should be provided using naming and addressing capability through object identification.

o Protection of object (including right management)

o Connecting to anything using object identification

o Service and location discovery

Therefore, identity protocol for object, i.e., HIP extensions, should be developed in order to perform mapping/binding capability and support the features required in communications between objects.

4. HIP Architecture for Object to Object Communications

4.1. The mapping relationships between host and object(s)

4.1.1. Host = Object (one to one mapping)

In case of a host is equal to an object, there is one to one mapping relationship between host and object. Most of information devices such as PC, etc are included in this case.

For example, if you use a telephone device, the device as host can be allocated a telephone number as service ID and be treated the same object.

4.1.2. Host =! Object (one to many mapping)

In case of a host is not equal to an object, there is one to many mapping relationship between host and object(s). Content server, RFID tags/Reader, etc are included in this case.

There are two kinds of one to many mapping as follows (see Figure 2):

- As shown in Figure 2 (a), host including objects such as content server, a host includes many objects and these objects should be identified using content ID, etc.

- As shown in Figure 2 (b), host with remote objects such as RFID tags, a host has many remote objects and these objects should be identified using RFID code, etc. In this case, each object might be non IP.

4.2. The stack architecture

The original stack architecture of HIP can be extended according to the mapping relationships between host and object(s).

- As shown in Figure 3 (a), objects in a host (case #1), the end point is the same with current HIP architecture. However, each object in service layer should be identified by a host using mapping protocol for object.

- As shown in Figure 3 (b), remote objects (case #2), the end point will be each object. This means that host location is different from end point(s). Thus, current HIP should be extended to support several end points with a host. From object information in service layer, each object identity should be defined.

Detailed protocol extensions will be specified in Section 5.

4.3. Object mapping schemes

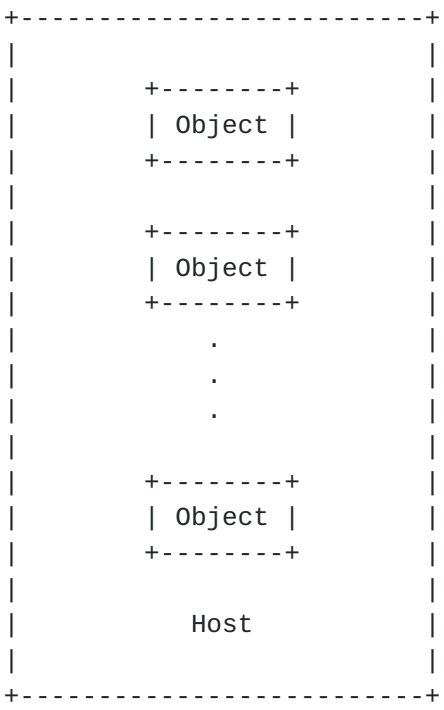
We can consider two kinds of object mapping schemes using one to many mapping relationship as follows:

- Direct mapping (Figure 3 (a))

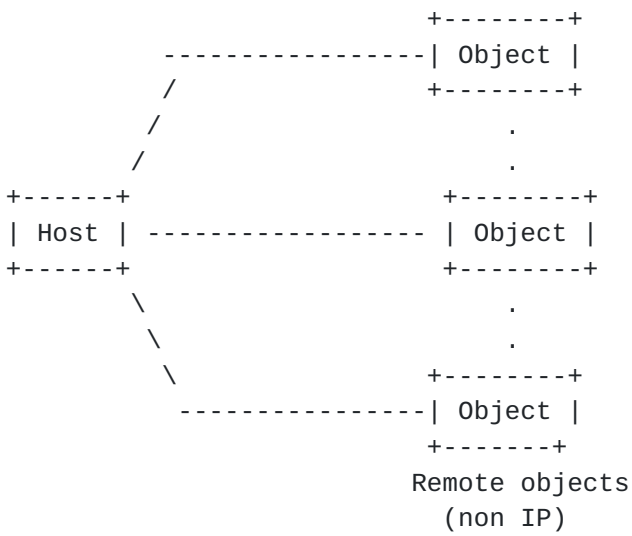
  An object at application layer is directly reachable to host entity at network attachment point which IP is terminated. An object is located on top of TCP/IP protocol stack. For example, host including objects such as content server, a host includes many objects and these objects should be identified using content ID, etc.

- Indirect mapping (Figure 3 (b))

  An object at application layer is remotely reachable through non-IP interface to host entity at network attachment point which IP is terminated. An object is located outside of physical network attachment which IP is terminated. For example, host with remote objects such as RFID tags, a host has many remote objects and these objects should be identified using RFID code, etc. In this case, each object might be non IP.

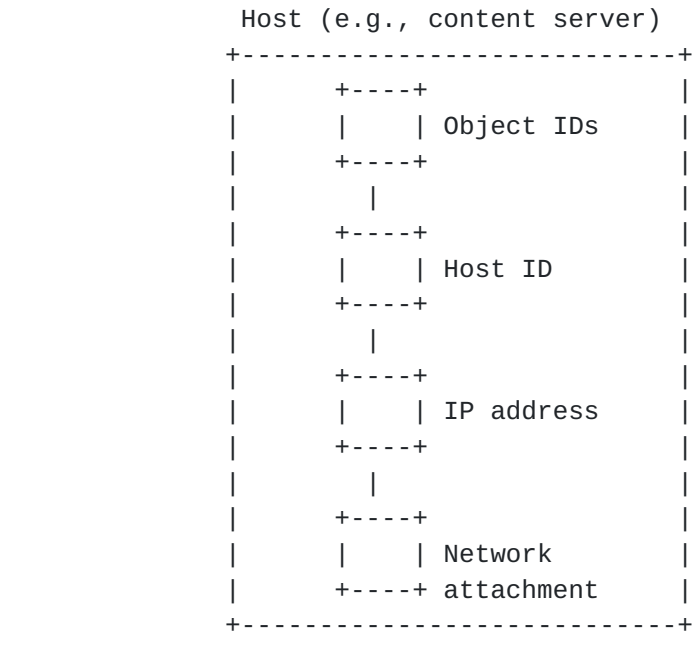


(a) Host including objects(e.g., content server)

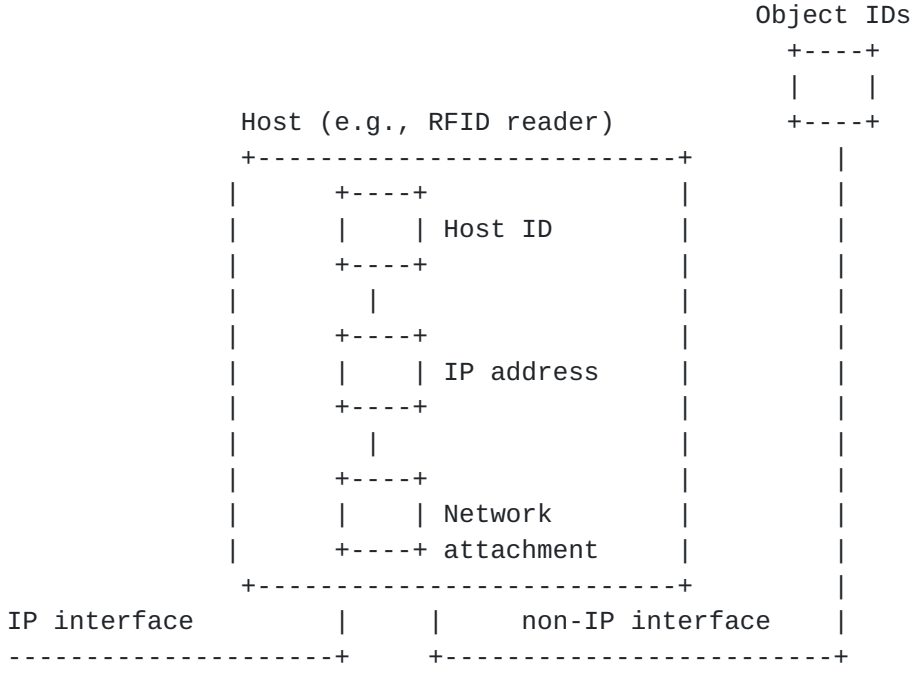


(b) Host with remote objects(e.g., RFID tags/Reader)

Figure 2 Mapping between host and objects (one to many mapping)



(a) Case #1: Objects in a host (host location = end points)



(b) Case #2: Remote objects (host location =! end points)

Figure 3 Extension of stack architecture

The proposed address and identifier mapping structure has the following advantages.

- o Perform two functions together - Routing using network prefix information and identification code using service IDs
- o Connecting to Anything - Provide the connectivity to end device without additional equipment such as Network Address Translator
- o Scalability - enough name space for supporting object-to-object communications
- o Security - security solution using HIP hash function, etc

5. HIP Extensions

5.1. Case #1: Objects in a host

In case of Figure 3 (a), several object identifiers as well as host identity should be delivered to each host for mapping information between host identity and object identities.

In order to deliver object information, this document newly defines a new TLV, i.e., Object_ID (see Section 5.3.).

5.2. Case #2: Remote objects

As case of Figure 3 (b), Object Identity (OI) information instead of host identity should be delivered to each host for mapping information between IP address and object identities.

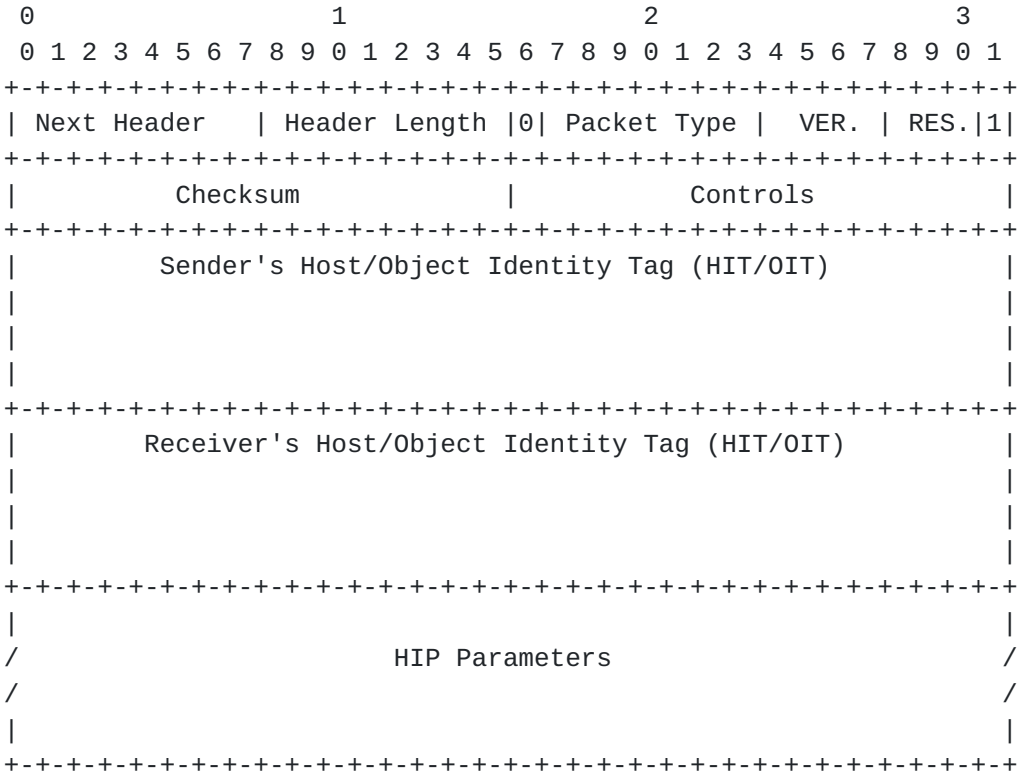
Thus, this document newly specifies Object Identity Tag (OIT) in HIP message. Each OIT typically identifies a service and can also identify end point.

5.3. Packet format

5.3.1. Proposal #1

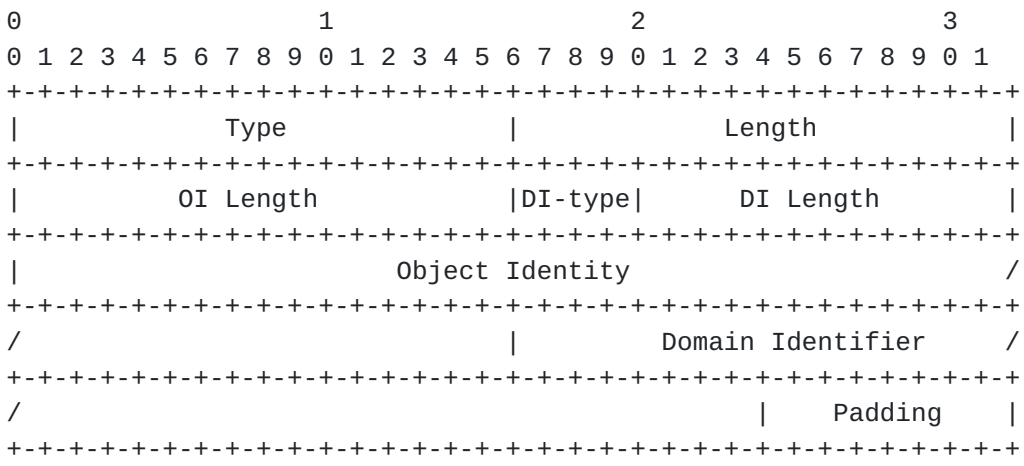
To support the previous extended architecture for object, the current HIP packet should be extended as follows.

o HIP header (include OIT)



The information for object should be included HIP header according to specific cases as described in Figure 3.

o Object_ID (newly defined from HOST_ID of HIP)



Type	TBD
Length	length in octets, excluding Type, Length, and Padding
OI Length	length of the Object Identity in octets
DI-type	type of the following Domain Identifier field
DI Length	length of the FQDN or NAI in octets
Object Identity	actual Object Identity
Domain Identifier	the identifier of the sender

The Object Identity is generated from Service IDs defined for specific applications/services. The detailed algorithms and formats follow the concept of the existing HIP specified in [RFC5201].

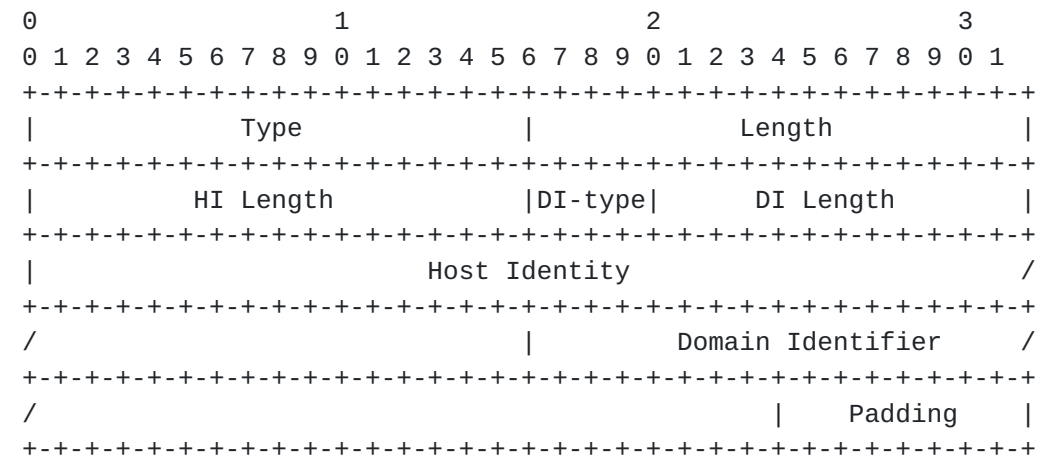
Other packet formats are subject to change according to HIP.

5.3.2. Proposal #2

For security association, there is an alternative to keep the existing Host_ID and add new Domain Identifier type for the object ID. In this case, we can use the existing HIP for security association.

Note: This is a result of hiprg e-mail discussion[4]. For this method, we need further discussion.

o HOST_ID



and add the following new DI-type:

The following DI-types have been defined:

	Type	Value
	none included	0
	FQDN	1
	NAI	2
+	Object ID	3

and then specify a new Domain Identifier format for the Object ID.

5.3.3. Comparison of two proposals

TBD

5.4. Protocol operations and procedures

HIP basic operation (an example)

- In case of communications suing RFID reader/tags, HIP Initiator can be a RFID reader which is connected to a RFID tag (object) using air interface and HIP Responder can be the information server which stores all information of RFID tags. And then, if this information server has a role of HIP rendezvous server, a client can get binding information between Host (HIP Initiator)and an object behind RFID reader for reachability to object(S) as end point(s).

- The RFID reader has one-to-many mapping relationship. So, a host identity of RFID reader maps onto many object identities.

- For IPsec security associations, HIP will definitely be terminated at the RFID reader because HIP should be tightly coupled with network layer. Similar with objects inside server, although each object is located remotely through air interface with RFID reader, we would like to consider RFID reader and tag as the same node virtually.

- In this case, we can consider two solutions.

    - The one is to put new name space (i.e., object identity) on top of HIP with RFID reader. This is the similar with case #1 in Figure 3 (a).

    - The other is that object identity replaces host identity on top of network layer of RFID reader as we originally suggested in case #2 in Figure 3 (b). However, if we keep the existing Host ID as we discussed in Section 5.3.2. proposal #2, this solution can't be applicable.

Protocol procedures

We illustrate the basic protocol procedure of sending a data packet to an object and mappings/bindings that are involved as shown in Figure 4:

- Find a node on which the required object resides. This requires finding object and end point through object ID registration. Name resolution using DNS is optionally required.

- Find a network attachment point to which the node is connected. This requires finding location. For this, a client gets binding information of object ID and IP address.

- Find a path from the client to object(s). The client can reachable to object(s) using routing path and binding information between HIP initiator and object(s). The datagram which is transferred to object(s) might have the information of object ID.

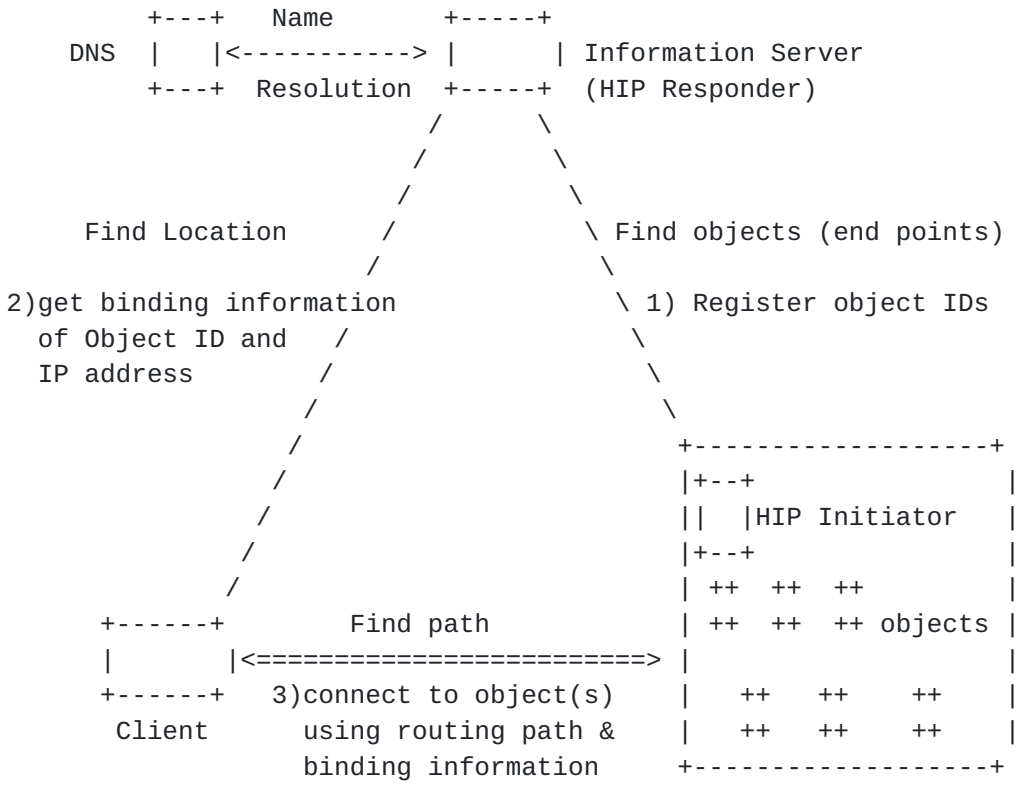


Figure 4 Protocol procedure for connecting objects

6. Considerations for HIP Extensions

6.1. Security association

It is critical to provide security association for secure binding between object identity and host identity [5]. For our cases, we can consider connection latching mechanism for IPsec channels [6].

6.2. Support of DNS, and HIP rendezvous server

In order to support from existing infrastructure, including DNS, and HIP rendezvous server, it is required to define DNS resource records. The newly defined DNS resource records should include information on object identifiers and object identity tags (OITs)

6.3. Protocol overhead

Real time communications and some limitation of power and packet size, lightweight identity handshake for datagram transactions is critical.

6.4. Common identifier for object

Most of identifiers for object specified with different format according to applications. However, in order to contain information of all objects in HIP message and interoperate globally, it is required to specify common identifier and rules to accommodate all objects with unified format.

6.5. Specific user cases

HIP for object can use original advantages of HIP for specific user cases.

- Identity-based roaming and mobility
- Hierarchical routing
- Addressing and location management
- Multi-homing
- Rendezvous service (or mechanism)
- DNS service

6.6. Services using extended HIP

The proposed extended HIP can provide an integrated solution for personal location and management through identification /naming /addressing including ID registration, location tracking, dynamic mobility control, and security using the following networking services:

- Identity management (IdM) services for the management of the identity life cycle of objects including managing unique IDs, attributes, credentials, entitlements to consistently enforce business and security policies.

- Location management services for real-time location tracking, monitoring, and information processing of moving objects similar with Supply Chain Management.

- Networked ID (N-ID) services for providing communication service which is triggered by an identification process started via reading an identifier from identifier storage such as RFID tag, barcode label, smartcard, etc.

- Home networking services for the management of multiple object identities in a host and/or remote host using RFID tag, ubiquitous sensor, etc.

7. Security Considerations

This document has specific security considerations as described in Section 6 and aligns with the security requirements in [RFC4423] and [RFC5201].

8. IANA Considerations

This document has no actions for IANA.

9. References

9.1. Normative References

None

9.2. Informative References

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

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

[Y.NGN-UbiNet] ITU-T TD280Rev1 (NGN-GSI), "Initial Draft Recommendation Y.NGN-UbiNet, Overview and Principles for Ubiquitous Networking in NGN", work in progress, September 2008.

[Y.IPv6-ID]ITU-T TD252 (NGN-GSI), "Initial Draft Recommendation Y.ipv6-object (Framework of Object Mapping using IPv6 in NGN)," work in progress, September 2008.

[1] Gyu Myoung Lee, Jun Kyun Choi, Taesoo Chung, Doug Montgomery, "Standardization for ubiquitous networking in IPv6-based NGN," ITU-T Kaleidoscope Event - Innovations in NGN, pp.351-357, May 2008.

[2] EPCglobal, "EPCglobal Object Name Service (ONS) 1.0.1," May 2008.

[3] Content ID Forum (cIDf), "cIDf Specification 2.0," April 2007.

[4] IETF HIP-RG mailing group discussion, available at https://listserv.cybertrust.com/pipermail/hipsec-rg/2008-December/000545.html

[5] Heer, Varjonen, "IP Certificates," IETF Internet-Draft, draft-ietf-hip-cert-00.txt, work in progress, October 2008.

[6] N. Williams, "IPsec Channels: Connection Latching," IETF Internet-Draft, draft-ietf-btns-connection-latching-08.txt, work in progress, November 2008.

Appendix A. Change History

Changes from November 2, 2008 version to March 9, 2009 version:

o Add Section 4.3. object mapping schemes

o Change Figure 3. Extension of stack architecture

- Add new proposal for protocol extension in Section 5.3
- Add Section 5.4. Protocol operations and procedures and Figure 4
- Add additional considerations in Section 6

Author's Addresses

Gyu Myoung Lee
Information and Communications University (ICU)
119 Munjiro, Yuseong-gu, Daejeon, 305-732, KOREA

Phone: +82-42-866-6828
Email: gmlee@icu.ac.kr

Jun Kyun Choi
Information and Communications University (ICU)
119 Munjiro, Yuseong-gu, Daejeon, 305-732, KOREA

Phone: +82-42-866-6226
Email: jkchoi@icu.ac.kr

Seng Kyoun Jo
Electronics and Telecommunications Research Institute (ETRI)
138 Gajeongno, Yuseong-gu, Daejeon, 305-700, KOREA

Phone: +82-42-860-6461
Email: skjo@etri.re.kr

Jeong Yun Kim
Electronics and Telecommunications Research Institute (ETRI)
138 Gajeongno, Yuseong-gu, Daejeon, 305-700, KOREA

Phone: +82-42-860-5311
Email: jykim@etri.re.kr

Acknowledgment

The authors wish to thank Tom Henderson for providing valuable input and comments in this document.