HIP Working Group
Internet Draft
Intended status: Informational
Expires: September 2010

Gyu Myoung Lee
TELECOM SudParis
Jun Kyun Choi
KAIST
Seng Kyoun Jo
Jeong Yun Kim
ETRI
Noel Crespi
TELECOM SudParis
March 8, 2010

Naming Architecture for Object to Object Communications
draft-lee-object-naming-02.txt

Status of this Memo

Abstract

This document explains the concept of object to object communications and describes naming issues for object identification. In order to develop protocols for object to object communications, this document provides the naming architecture according to mapping relationships between host and object(s). In addition, considerations of protocols for naming 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 one of new capabilities for future network will be the ubiquitous networking such as the Internet of things. This networking capability should support "Any Time, Any Where, Any Service, Any Network and Any Object (so-called "5-Any")" 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 types objects. However, there is no consideration for certain new type of objects (e.g., contents, RFID tags, sensors, etc) as end points.

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

For architectural aspect, this document shows architecture for identity processing and mapping relationship between several identities with conceptual diagram for providing connectivity to objects.

According to several alternative architectures for object naming, this document aims to provide requirements and right direction for protocol development for realization of object to object communications.

2. Object to Object Communications

2.1. Definition of object

object: a model of an entity. An object is characterized by its behavior. An object is distinct from any other object. An object interacts with its environment including other objects at its interaction points. An object is informally said to perform functions and offer services (an object which performs a function available to other entities and/or objects is said to offer a service). For modeling purposes, these functions and services are specified in terms of the behavior of the object and of its interfaces. An object can perform more than one function. A function can be performed by the cooperation of several objects.

NOTE - Objects include terminal devices (e.g. used by a person to access the network such as mobile phones, personal computers, etc), remote monitoring devices (e.g. cameras, sensors, etc), information devices (e.g. content delivery server), products, contents, and resources.

NOTE - the above definition was quoted from ITU-T [Y.2002].

2.2. Concept of object to object communications

For ubiquitous networking [Y.2002], 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 support "Any Time, Any Where, Any Service, Any Network and Any Object" 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.

2.3. Various types of objects

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, Smartphones 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.

The object means that the user or other entity which is connected to the network. It includes almost everything around us such as remote monitoring and information device/machine/content, etc.

Figure 1 shows the connection of Internet with the relationship between humans and objects in terms of identification and location in specifically mobile environments. The types of objects on the end-user side include the following: personal devices, information devices, RFID/sensors, contents, appliances, vehicles, etc.

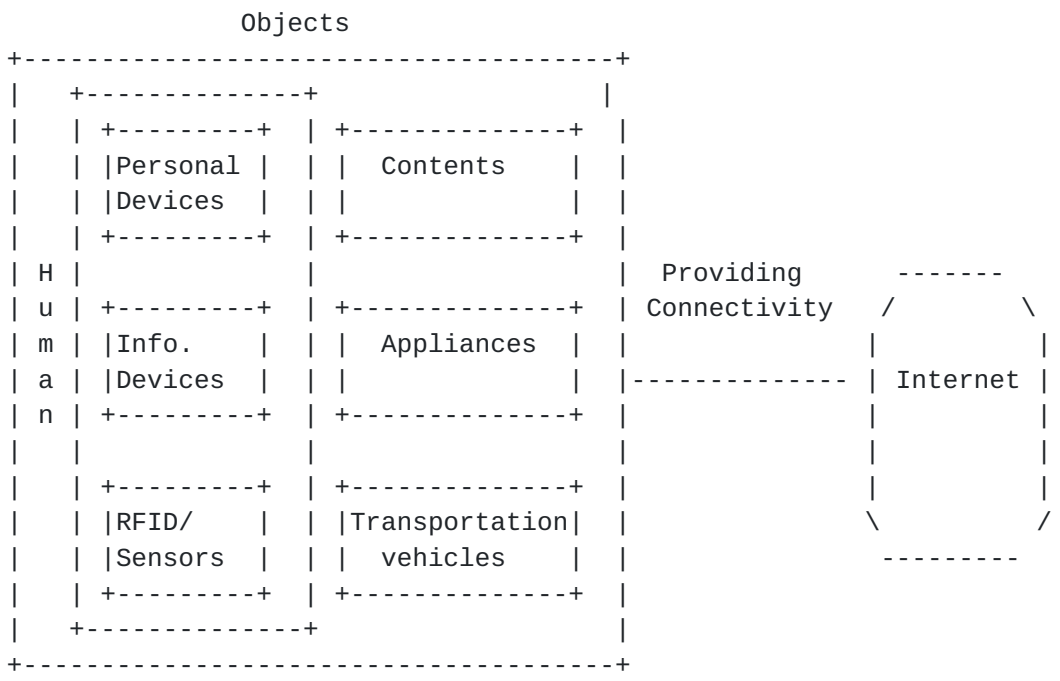


Figure 1 Communications with objects through Internet

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.

- Services (i.e., information related to applications/services)
- End points (i.e., global unique identifier)
- Location (i.e., IP address)
- 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. An identifier is capable of identifying all objects and facilitates objects-to-objects communications. In particular, a 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 numbering plan, 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.

- Object IDs: include RFID, Content ID, telephone number, URL/URI, etc
- Communication IDs: include session/protocol ID, IP address, MAC address, etc

In this document basically consider an "Object ID" which generally takes the form of an application-specific integer or pointer that uniquely identifies an object.

3.3. Examples of 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 using object identification

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

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

- Scalability with enough name space to support new devices/machines enabling communications
- Protection of object (including right management) using security function
- Connecting to anything for providing the connectivity to end device without additional equipment such as Network Address Translator using object identification
- Service and location discovery through performing two functions; Routing using network prefix information and identification code using object IDs

4. Naming Architecture for Objects

4.1. Layered architecture for identity processing

As shown in Figure 2, the layered architecture of identity processing requires specific processing capabilities at each layer. Each user/object in applications 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 object IDs through identification and authorization. Each object ID also should be associated with communication IDs through mapping/binding [Y.ipv6-ID].

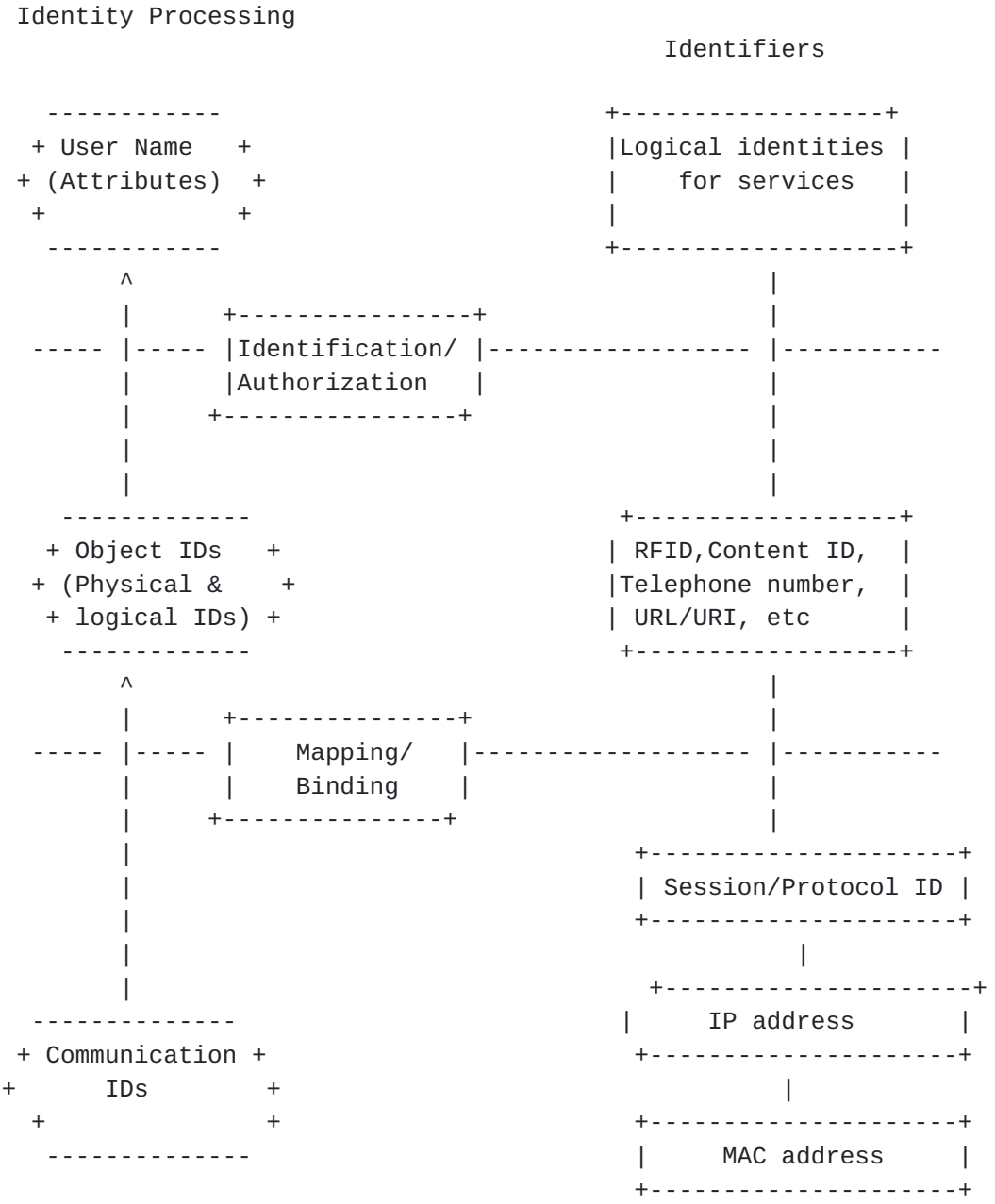


Figure 2 Layered architecture for identity processing

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

In this document, host means a device that communicates using the Internet protocols (i.e., IP addresses).

4.2.1. 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.

Current HIP has been typically designed for this case.

4.2.2. 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 3):

- As shown in Figure 3 (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 3 (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.

4.3. 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 4 (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 4 (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.

4.4. Object mapping schemes

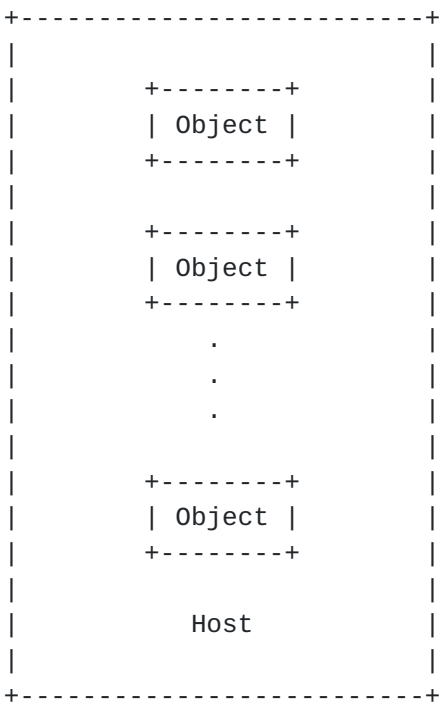
There are two kinds of object mapping schemes using one to many mapping relationship as follows:

- Direct mapping (Figure 4 (a))

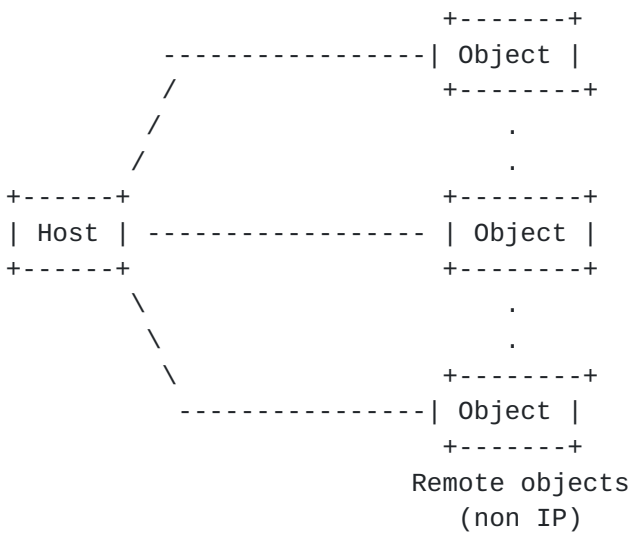
  An object at application layer is directly reachable to host entity at network attachment point which IP is terminated. An object is located on top of TCP/IP protocol stack. For example, host including objects such as content server, a host includes many objects and these objects should be identified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

- Indirect mapping (Figure 4 (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.

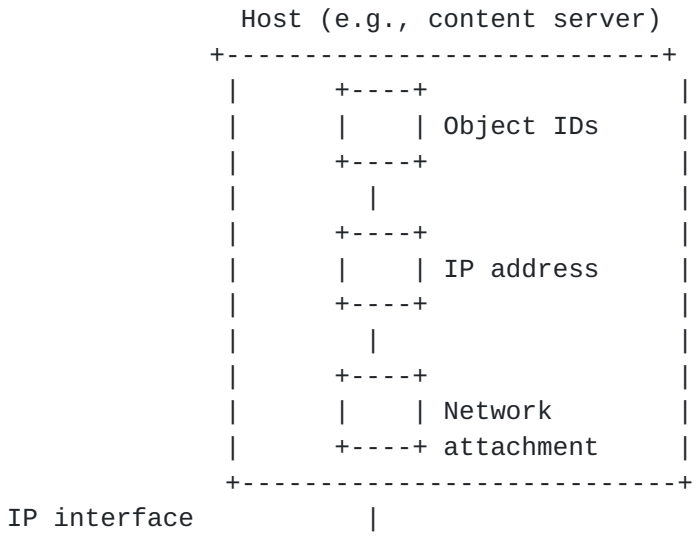


(a) Host including objects(e.g., content server)

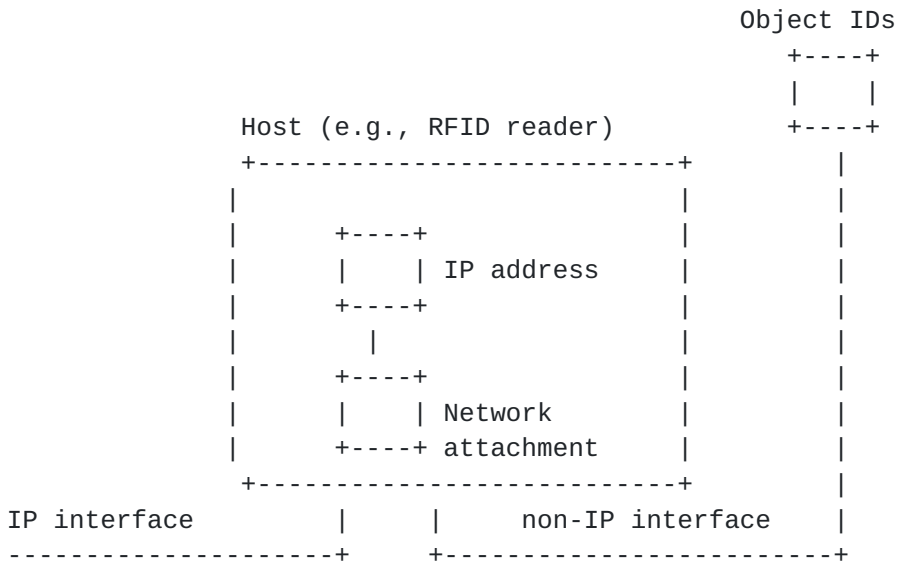


(b) Host with remote objects(e.g., RFID tags/Reader)

Figure 3 Mapping between host (IP address) and objects (object IDs)
(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 4 Extension of stack architecture

4.5. Providing connectivity to objects

For providing connectivity to objects using object identification, the Figure 5 shows object mapping/ binding with IP address for IP connectivity to all objects in end-user side. This scheme can provide the global connectivity with Internet to objects through the association (e.g., mapping/binding) between identifier for object and IP address.

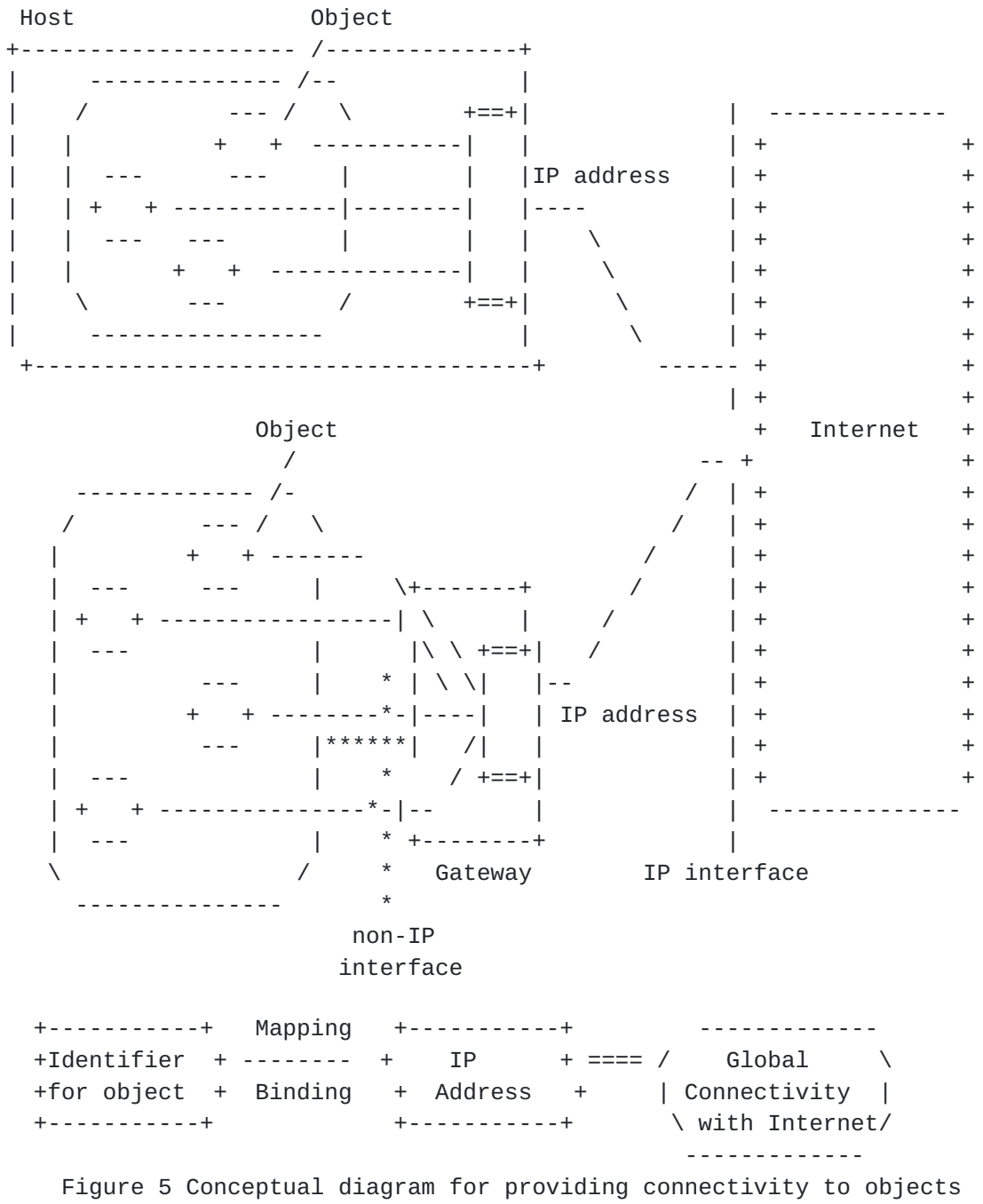


Figure 5 Conceptual diagram for providing connectivity to objects

5. Considerations of Protocols for Naming Objects

5.1. Security association

It is critical to provide security association for secure binding between object identity and IP address similar with HIP [5].

5.2. Support of DNS

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.

In order to support from existing infrastructure, including DNS, it is required to define DNS resource records. The newly defined DNS resource records should include information on object IDs.

5.3. Protocol overhead

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

5.4. 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 protocol message and interoperate globally, it is required to specify common identifier and rules to accommodate all objects with unified format.

5.5. Relationship with locator for mobile object

As the location of object(s) is frequently changed in mobile environment, the information of object ID should be resolved with the information of location. It is required to use the concept of ID/Locator separation considering object ID.

5.6. Specific user cases

Naming protocol 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

5.7. Services using naming objects

The proposed naming objects 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.

6. Security Considerations

This document has specific security considerations as described in Section 5 and aligns with the security requirements in [RFC4423] and [RFC5201].

7. IANA Considerations

This document has no actions for IANA.

8. References

8.1. Normative References

None

8.2. 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.2002] ITU-T Y.2002, "Overview of ubiquitous networking and of its support in NGN", November 2009.

[Y.IPv6-object]ITU-T TD43 (WP5/13), "Framework of Object Mapping using IPv6 in NGN", work in progress, September 2009.

[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, "HIP Certificates," IETF Internet-Draft, draft-ietf-hip-cert-02.txt, work in progress, October 2009.

Author's Addresses

Gyu Myoung Lee
Institut TELECOM, TELECOM SudParis
9 rue Charles Fourier, 91011, Evry, France

Phone: +33 (0)1 60 76 41 19
Email: gmlee@it-sudparis.eu

Jun Kyun Choi
Korea Advanced Institute of Science and Technology (KAIST)
119 Munjiro, Yuseong-gu, Daejeon, 305-732, KOREA

Phone: +82-42-350-6122
Email: jkchoi@ee.kaist.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

Noel Crespi
Institut TELECOM, TELECOM SudParis
9 rue Charles Fourier, 91011, Evry, France

Phone: +33 (0)1 60 76 46 23
Email: noel.crespi@it-sudparis.eu